National Library of Canada
Bibliothèque nationale du Canada

Canadian Theses Service
Services des thèses canadiennes

Ottawa, Canada
K1A 0N4

CANADIAN THESES

THÈSES CANADIENNES

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30.

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30.

THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED

LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE

Canada

NL-339 (r.86/06)

The University of Alberta

THE PLA FOLDING PROBLEM

by

© Darrell D. Makarenko

A thesis
submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree
of Doctor of Philosophy

Department of Computing Science

Edmonton, Alberta
Spring, 1986

Permission has been granted to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film.

The author (copyright owner) has reserved other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without his/her written permission.

L'autorisation a été accordée à la Bibliothèque nationale du Canada de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur (titulaire du droit d'auteur) se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation écrite.

ISBN 0-315-30104-X

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: Darrell D. Makarenko

TITLE OF THESIS: The PLA Folding Problem

DEGREE : Doctor of Philosophy

YEAR THIS DEGREE GRANTED: 1986

Permission is hereby granted to The University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(Signed) Darrell Makarenko .....

Permanent Address:

617 Coventry Road
Winnipeg, Manitoba
Canada

Dated December 2, 1985

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled **The PLA Folding Problem** submitted by **Darrell D. Makarenko** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy.**

.....
Supervisor

.....

.....

.....

.....

Date: December 9, 1985

for my parents

ABSTRACT

PLA folding is a technique used to minimize the computer chip area requirements of Programmable Logic Arrays. This thesis focuses on the fundamental properties of the theoretical problem based on this technique. Current knowledge of the PLA folding problem is limited to empirical evidence gained from studies using heuristic methods. There is little understanding of the basic nature of the problem. This thesis provides some fundamental theoretical results about the PLA folding problem along with some empirical measurements for randomly generated PLAs.

First, a universal PLA model is provided, one which has both a topological component for describing structural properties of PLAs and a set-theoretic component for analytically representing the folding problem. A folding taxonomy is developed from the model, and used to classify the properties of PLAs, folding methods and folding algorithms.

A new branch and bound algorithm is described for finding optimal folding sets. A new performance measure is also introduced for comparing the relative effectiveness of the various optimal folding algorithms. The derivation of the measure is from empirical data obtained by folding randomly generated PLAs.

Using a random selection heuristic as a basis, a probability density function is derived for expected number of folds of random PLAs. This is the major contribution of the thesis. The analytical derivation of this result illustrates the interactions that occur between the basic PLA properties in the process of folding. This PDF is also used in the derivation of a new

folding heuristic. It is the first such heuristic to have an analytical basis and is shown empirically to perform better than the others available. By applying the heuristic iteratively, it is possible to achieve results arbitrarily close to those of optimal folding.

Finally, a new performance measure is introduced for comparing heuristic folding algorithms. This has become necessary because existing folding algorithms have no theoretical or analytical basis. The limited empirical evidence provided by researchers in the field is insufficient and the effectiveness of many methods remains in doubt. Empirical results presented here will demonstrate that this doubt is, in many cases, fully justified.

The theoretical results of this thesis produce a clear understanding of the nature of the PLA folding problem. The exact relationship between foldability and the fundamental characteristics of a PLA are explained. Empirical data are provided to demonstrate the effectiveness of the analytical derivations in modeling the folding problem for randomly generated PLAs.

Acknowledgements

I welcome this opportunity to acknowledge the efforts of all those who have helped me in completing this work.

First, I would like to express my sincere thanks to my supervisor, Dr. John Tartar, for providing both encouragement and advice throughout the course of my studies. The numerous occasions on which he has made an extra effort on my behalf are greatly appreciated.

I would like to thank Dr. Duane Szafron for his constant encouragement and the many hours that he spent with me reviewing mathematical formulae. His honest concern and his willingness to listen have been invaluable to me.

I greatly appreciate the efforts of Dr. Jon Muzio for his careful reading of the thesis, his genuine interest in my work, and for going out of his way to provide assistance when needed.

I would like to thank Ken Hruday for some interesting discussions on PLA folding, as well as all of my fellow graduate students and the faculty and staff of the Department of Computing Science for providing an atmosphere of friendship and academic curiosity.

Both the Natural Sciences and Engineering Research Council of Canada and the Province of Alberta Heritage Trust Fund are acknowledged for their financial support of this work.

Finally, I would like to express my deepest appreciation to Lynne, for her continual love and support.

Table of Contents

Chapter	Page
Chapter 1: Introduction	1
Chapter 2: Tutorial and Critical Review of the Literature	5
Chapter 3: PLA Model	16
3.1. Topological Model	17
3.2. Set-Theoretic Model	23
Chapter 4: PLA Folding Taxonomy	30
4.1. Architectural Classification Scheme	31
4.2. Classification of Folding Techniques	33
4.3. Categorizing Folding Algorithms	40
4.4. Summary and Application	41
Chapter 5: Optimal PLA Folding	44
5.1. Size of the Problem	45
5.2. New Branch and Bound Algorithm	48
5.3. Empirical Data	55
5.4. Bounding Measures	59
Chapter 6: A Theoretical Analysis of PLA Folding	63
6.1. Basic Definitions	64
6.2. Deriving P_c	65
6.3. Expected Number of Folds	73
6.4. Row orders - Restriction #3	79
6.5. Experimental Data	89
6.6. Removing the Evenly Distributed Density Assumption	96
6.6.1. Revising P_c	96
6.6.2. Revising Row Orders	98
Chapter 7: The Extended Random Selection Heuristic	104
7.1. Analytical Derivation	105
7.2. Empirical Data	109
Chapter 8: Heuristic Comparison Measures	114
Chapter 9: Conclusions	117
References	120

List of Tables

Table		Page
4.1	Summary of Taxonomy	41
4.2	Application of Taxonomy to Current Literature	43
5.1	Counting distinct folding sets	47
5.2	Bounding Measures for PLAs of size r=20, c=14	62
8.1	Heuristic Comparison Measures	115

List of Figures

Figure	Page
2.1 - Example PLA	8
2.2 - Simple Column Folding	9
2.3 - Simple Row Folding	9
2.4 - Multiple Row Column Folding	10
2.5 - Bipartite Folding	13
3.1 A Nor-line	19
3.2 A Simple Plane	20
3.3 A PLA - Two Interconnected Planes	21
3.4 Wire Segments	22
3.5 Column Folding of a Simple Plane	23
3.6 A Example Single Plane PLA	24
3.7 Ordered Partition of a PLA	28
4.1 Folding of Open Ended Wires	35
4.2 Folding Closed Ended Wires	35
4.3 External Folding of a Two Plane PLA	37
4.4 Internal Folding of a Three Plane PLA	37
5.1 Optimal Folding Size vs Density	56
5.2 Optimal Folding Size vs. # of Columns	57
6.1 Evenly Dense PLA	65
6.2 P_c vs density for r=20	68
6.3 P_c vs r for density=20%	68
6.4 P_c and $P_c approx$ vs density for r=20	71
6.5 Row Orders Implied by Previous Folds	80
6.6 PDF(n,r,c,d) vs n as derived	88
6.7 P_c derived and P_c measured vs d	90
6.8 Q(n) derived and Q(n) measured	91
6.9 PDF derived and PDF measured	91
6.10 PDF for Random Selection Heuristic and Optimal Folding	94
7.1 $pdf_E(n)$ for K=2 and K=10	108
7.2 Average Folding Size vs K as derived from $pdf_E(n)$	109
7.3 $pdf_E(n)$ for K=10 as derived and measured	109
7.4 Average Folding Size vs K as derived and measured from $pdf_E(n)$	109

Chapter 1

Introduction

In the past few years there have been significant advances made in the technology for fabricating silicon devices. Continuous improvements in the size, speed and power consumption of transistors now allow designers to place upwards of 100,000 components on a single silicon chip. This has not been without difficulty however, and the challenge remaining is to make use of this technology in the form of useful products. Clearly, the bottleneck to this goal is in the design phase. How does one coordinate the design of a device composed of 100,000 transistors within a time frame smaller than the period of obsolescence of the product itself? In addition, new standards of reliability and robustness are expected by the consumers of modern electronic devices. It is clear that the organizational procedures used in the past are no longer effective for producing modern complex devices.

One approach used to overcome this problem has been the development of architectural structures that are not necessarily the most efficient in terms of chip area but are efficient in terms of design effort. The Programmable Logic Array (PLA) is such a structure and has gained much popularity since the maturing of VLSI technology.

The major advantages of PLAs is that they can dramatically reduce the design effort required to implement combinational and sequential logic. They are particularly useful in the design of control logic for microprocessors. By using PLAs, one can completely automate much of the design process and thus reduce significantly the design time for logic devices. This also results

in an increase in reliability.

However, industrial PLAs tend to be sparse and thus wasteful of chip area. Silicon area is a valuable resource and so to make the design time - chip area tradeoff more favorable for PLAs, a technique known as folding is used to reduce the required chip area. Folding involves arranging the various inputs and outputs of the PLA so that they can share rows and columns within the PLA. The problem of finding the best arrangement that yields the most area reduction is known as the PLA Folding Problem.

This thesis is a theoretical study of the PLA folding problem. It is not a study of the properties of PLAs used in industrial applications however some of the results provided here may be of practical use. The current PLA Folding literature is lacking in theoretical work. Many of the ideas and algorithms presented are based only on an "intuitive" argument. The author could not find any work done on the nature of the problem itself in terms of determining how the characteristics of a PLA affect its foldability. After providing a brief tutorial on the problem and a complete review and critique of the current PLA folding literature, this thesis presents several such fundamental theoretical contributions to the available literature.

PLA folding comes in many different forms that are often dependent on constraints determined by the technology in which the PLA is to be implemented. Sometimes there are artificial constraints also imposed that may not pertain to any practical considerations. To allow comparisons among the works of various authors in the field, a standard form of PLA representation is needed. A PLA model, suitable for such a purpose, is described. It is a set-theoretic model and can represent all types of PLA

folding in an easily understandable manner.

Using this model it is possible to divide the current PLA folding techniques into several categories. A PLA folding taxonomy that is based on these categories is developed. Included in the taxonomy is a classification schema for the different types of algorithms available for folding PLAs. Some representative contributions in the current PLA folding literature are then classified according to this taxonomy.

As part of this work, an examination is done on the problem of finding the optimal arrangement of rows and columns of a PLA that results in the absolute highest savings in chip area. The problem has been shown to be np-complete and so no easy solution is possible. A new branch and bound algorithm is introduced and described. This algorithm is suitable as a practical tool for folding small to medium sized PLAs. In addition, it is used to illustrate some of the folding characteristics of PLAs. A number of theoretical results are presented and are used to derive two comparison measures. The first measure represents the difficulty in finding an optimal folding set for various PLA classes while the second measure represents the effective performance of various folding algorithms.

The most significant contribution of this research is a theoretical derivation of the expected folding of random PLAs. This derivation is based on a random selection heuristic and is in the form of a probability density function (PDF) in terms of fundamental characteristics of the PLA; r - the number of rows, c - the number of columns, d - the density, and h(x) - a function representing the distribution of the density among the columns. Empirical data are also provided to illustrate the effectiveness of these

formulae in modeling the PLA folding problem. The PDF is further developed into a heuristic algorithm for PLA folding. It is the first folding heuristic to have an analytical basis for expected results. Empirical data are provided to show that the heuristic works better than other available algorithms. Some interesting theoretical properties are observed.

To aid in evaluating the new heuristic, a heuristic comparison measure is developed and shown to be effective in comparing the performances of the different folding heuristics. Several of the popular heuristics are shown to perform poorly in folding PLAs.

Finally, as a conclusion to this work, this thesis is shown to have provided some fundamental contributions to the PLA folding problem. The results provided improve considerably our understanding of the nature of the problem. They provide a good framework upon which future work can be done and some interesting ideas for future research are included.

Chapter 2

Tutorial and Critical Review of the Literature

Before presenting the major contributions of this research, it is necessary to give an overview of the literature on PLAs and PLA folding. First the entire subject area of PLAs is reviewed including various minimization and optimization techniques. Next a brief tutorial on the specific area of PLA folding is provided. Finally a critical analysis of the current literature on PLA folding is presented and used as a basis for the contributions described in future chapters.

The use of Programmable Logic Arrays to implement boolean functions has been a technique well known in the literature for many years. Much of the early work was done by IBM in the early 1970s [FlM75, HCO74, Jon75, Loa75, Wei79]. Even at this early stage it was clear that PLAs were a useful tool for logic device design. The benefits of PLAs as described at that time [FlM75] are the same reasons that PLAs are in use today. Although no algorithms were available at the time, the techniques of partitioning logic designs into a number of separate PLAs and of decoding inputs to reduce the complexity and area of the PLA were well understood. However, no mention of PLA folding occurs in these early papers.

In the following few years, IBM continued to develop its work [GLL80, Sch80]. A lot of interest had been generated by MINI [HCO74] and other researchers began to get involved with logic minimization for PLAs [ArB78, Kam79, Rot78]. Logic minimization techniques advanced quickly because it was not a new area peculiar to PLAs. Much of the fundamental

work on sum-of-products reduction had already been examined in previous research to reduce chip counts on printed circuit boards.

It was around 1980 that PLA research began to expand significantly. This can be attributed primarily to the maturing of VLSI technology. PLAs were now seen as a feasible implementation for logic devices. The ideas of silicon compilation and automated design were becoming familiar [Ayr83] and PLAs were well suited to these applications. The Mead and Conway text [MeC80] introduced a large number of university faculty and graduate students to VLSI and to PLAs. Many new researchers entered the field and specific sub-areas were formed. It was now common to see results of PLA research being presented at major conferences on VLSI and Design Automation. These contributions came from both industry and academia.

Today PLA research can be subclassified into several distinct areas. For example, logic minimization is still an active area of PLA research. One of the popular reduction tools used by many is PRESTO. Douglas Brown describes an extended version of PRESTO [Bro81]. Another well know minimization program is PRONTO [MaP83]. There is little evidence provided to support the authors' claim that it works better than PRESTO. Both PRESTO and PRONTO are heuristic algorithms which perform near-optimal logic reduction.

The INTEL Corporation has produced a logic minimizer known as LOGMIN [TeW82]. Like PRESTO it is designed to generate logic equations from a finite state machine description. However, LOGMIN is not heuristic. It finds all solutions and thus is slower than the heuristic algorithms.

In addition to these implemented algorithms, there are also some theoretical contributions. In the work by Sasao [Sas81], the decomposition of multivalued boolean functions is examined and the optimal assignment of input variables and output phase is analyzed [Sas84] .

Throughout the work on logic minimization there are numerous discussions on the idea of automatically generating the logic equations of a PLA from some higher level description. In a number of papers [Bro81, MAP84, MSV83], the equations are generated from a finite state machine description and in others [GrN83, WiS83], a hardware description language is used. There are several good descriptions of the current state of design synthesis [Hia84, San85].

Another distinct area of PLA research is that of partitioning logic equations into separate PLAs to reduce the total area used. There are some recent results in this area [Hen83, Kan81, MiS83a, PuL84] The area of PLA testing and fault tolerance has also been the subject of recent interest [Pat80, SGM83, TFB82], and there are also some complete PLA systems that exist which combine all of the different research areas. RCA has produced APSS [Sta82, Sta83] to generate PLA layout masks from a boolean expression and Stanford has produced APLAS [KaC81, Kan81]. There are a variety of other working systems [MAP84, TeW82].

One of the novel ideas seen in the literature is that of Storage Logic Arrays. In an SLA, flip flops are located internal to the PLA and are used to implement finite state machines. A number of good references to SLAs are available [Goa81, LTM82, Pat80, PaW79, Smi82].

PLA folding is a technique that is relatively new in the literature. One of the first references to it is in the work by Wood [Woo79], where it is presented in a form slightly different from what is used today. The following is a brief tutorial on the PLA folding problem. A more detailed description is given in Chapter 3 of this thesis.

PLAs are an architectural structure used to implement AND-OR logic on a computer chip. As the focus of this thesis is the theoretical properties of the PLA Folding Problem, PLAs will be represented only in terms of the locations of active intersections. Active intersections correspond to locations of transistors and there are a number of good books available describing the technological realization of PLAs. Figure 2.1 shows a typical PLA with the AND-plane and the OR-plane indicated. The AND-plane calculates the minterms based upon the inputs and the OR-plane calculates the output functions based upon the minterms. The inputs in Figure 2.1 are labeled 1 to 5 and the outputs are 6 and 7. There are 5 minterms labeled m1 to m5. This PLA implements the two boolean functions

6 = (3 and 5) or (2 and 3 and 4) or (4)

7 = (1 and 4) or (1 and 2)

The area of the PLA can be represented as $r \times c$ where r is the number of rows and c is the number of columns. In this example the area is $5 \times 7 = 35$. It can be seen that many of the intersections contain no Xs because they are inactive. The density of the PLA, d, is the percentage of intersections that are active. Often, most of the intersections are inactive, representaing a significant waste of chip area. The purpose of PLA folding is

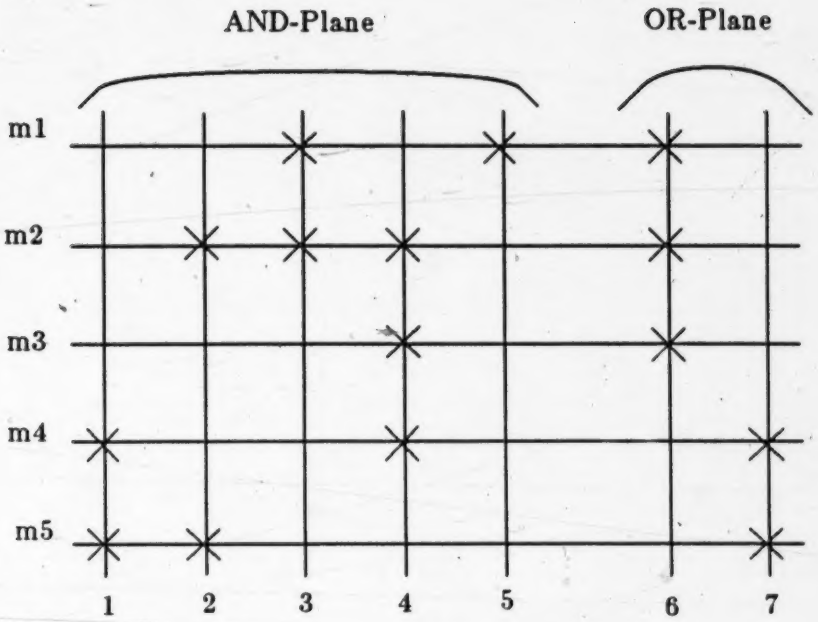

Figure 2.1 - Example PLA

to reduce this wasted space. The technique is based upon the observation that some of the inputs and outputs use completely disjoint sets of minterms and thus can be combined in a single column. Figure 2.2 illustrates this procedure in a form known as simple column folding. The area of the PLA has been reduced to $4 \times 5 = 20$ which represents a 42% savings over the previously shown unfolded version.

One can arbitrarily decide to fold the rows instead of the columns. Figure 2.3 illustrates simple row folding of the same PLA. The area of this PLA is $3 \times 7 = 21$.

The problem of deciding which inputs, outputs or minterms are to share rows or columns is known as the PLA folding problem. The objective is to find the arrangement which yields the greatest reduction in area. The

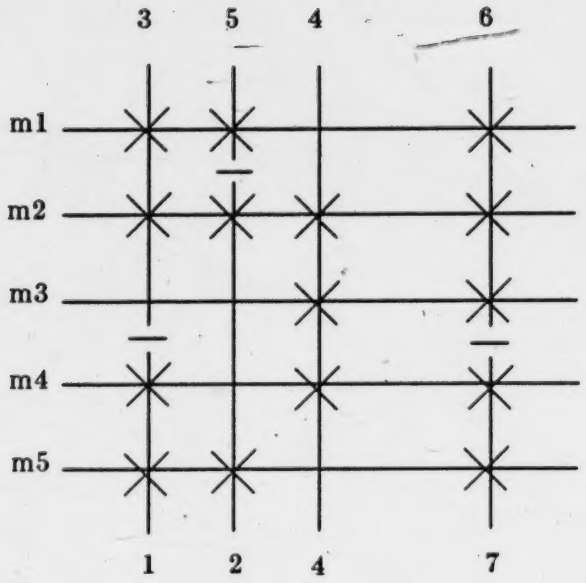

Figure 2.2 - Simple Column Folding

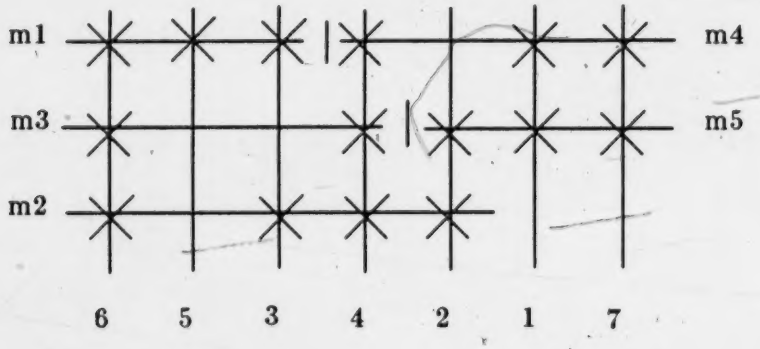

Figure 2.3 - Simple Row Folding

problem becomes more difficult if one allows both row and column folding within the same PLA as shown in Figure 2.4 If three or more inputs are allowed to share a single column then it is known as multiple folding.

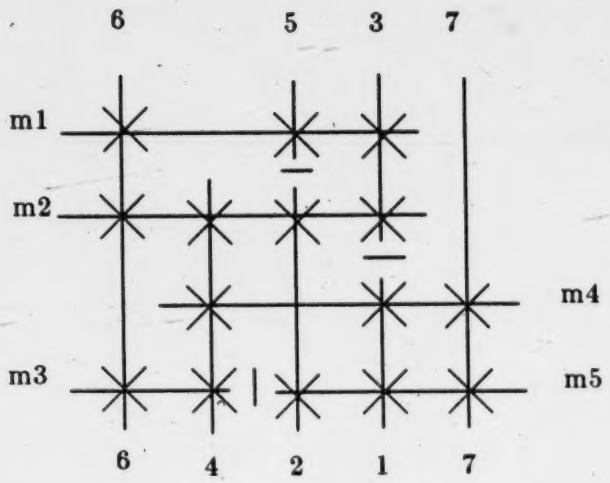

Figure 2.4 - Multiple Row Column Folding

The first mathematical formulation of the PLA folding problem was given by Hachtel et al in 1980 [HSN80]. In the paper the authors present, for simple column folding, a graph theoretic interpretation of the problem which has been almost exclusively used in the literature. Two columns are said to be adjacent or intersect if they have active intersections in any common rows. When two columns are folded they form a *folding pair* and the complete folding of a PLA is represented by a set of folding pairs known as a *folding set*. The interpretation is based on the construction of a column intersection graph upon which is superimposed a directed graph representing the set of column folding pairs. This model allows the authors to introduce a heuristic algorithm for finding a near optimal folding set. The algorithm involves a directed search with no backtracking. The heuristic that is used selects two columns, one with minimum adjacency to other columns and one with maximum adjacency. This heuristic is somewhat arbitrary and only a few

PLA examples are given as evidence to its effectiveness. The intuitive argument that the authors provide to justify the heuristic is weak. However this is the first PLA folding heuristic presented in the literature. The authors also provide an argument indicating that the problem is likely to be np-hard.

In 1982 a number of authors follow up on this early work with some new results. Luby et al [LVV82] prove that the problem is np-complete. As well, the authors present the problem as a variation of the bandwidth problem and suggest that techniques used for solving the bandwidth problem could be applied to the PLA folding problem. This idea appears impractical and has never been attempted.

In [HNS82a], Hachtel et al formalized a classification of the various PLA folding techniques. The terms "simple column folding", "simple row folding" and "multiple row column folding" were introduced. A new algorithm for performing row folding after column folding is also presented in the paper. The best description of all of the contributions by Hachtel et al can be found in [HNS82b]. Despite its title, "An Algorithm for Optimal PLA Folding", the paper describes a heuristic algorithm for near-optimal folding. It is the same algorithm as in the earlier papers, but is described in sufficient detail to allow others to implement it.

This original work by Luby, Hachtel and others laid the groundwork for many of the current researchers to enter the field. However, one problem with the heuristic algorithm that they introduced is that it is somewhat arbitrary. It does not have either an analytical or empirical basis for its performance. The intuitive argument that is given is not convincing and no measure for its effectiveness is provided. In the various papers, often just a

few PLA examples are provided.

Recently De Micheli and Sangiovanni-Vincentelli have expanded on their work considerably. A working software system called PLEASURE has been implemented and is based on their previously described heuristics. There are a number of good descriptions of the PLEASURE system [MiS83b, MiS83c, MiS83]. It handles both simple and multiple folding as well as outside constraints placed upon the rows and columns. Despite the enhancements, no new arguments are given as to the effectiveness of the heuristic.

At the same time that PLEASURE was developed, work was being done by others on a specific type of folding called *bipartite* folding. In bipartite folding, all of the column folds must occur between the same rows for all column pairs. Figure 2.5 shows an example of bipartite column folding with all folds occurring between minterms m3 and m4. This is the same PLA as in the previous figures. Note that with the added constraint of bipartite folding, less folding is possible. The best analysis of this is by Egan and Liu [EgL82, MiS83]. Kuo et al have produced some interesting results in [Hu83, KCH85]. The problem with bipartite folding is that it is an artificial constraint. There is not any technological reason why all the folds should occur at the same row level.

Very little work has been done on the problem of finding optimal solutions to the PLA folding problem since it was shown to be np-complete. There are a number of branch and bound algorithms available for this problem [Gra82, LeL84] and a new one is presented as part of this thesis. There is no method for comparing the relative performance of these different

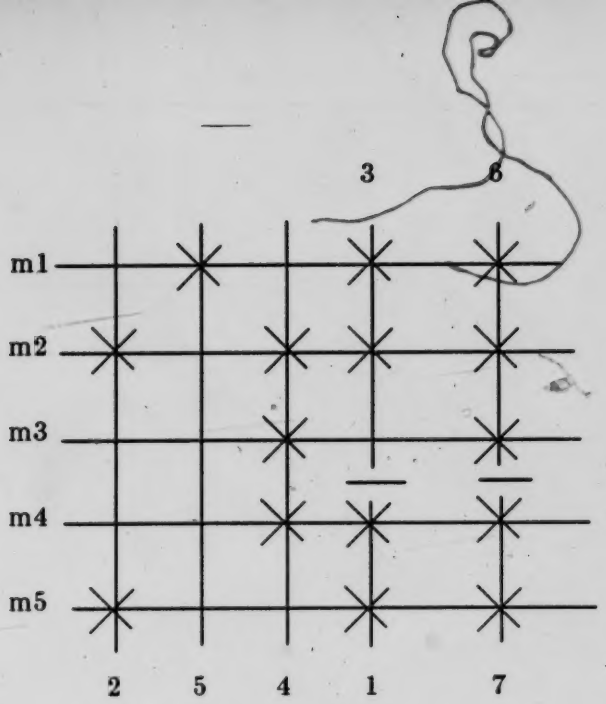

Figure 2.5 - Bipartite Folding

optimal folding algorithms. This thesis will introduce a comparison measure for such a purpose.

There are a few other varied contributions in the field of PLA folding. Suwa [SuK81, Suw82] presents a complete PLA system and Pailloton [Pai81] presents an algorithm for multiple folding. Pailloton's paper helps to illustrate some of the problems in the PLA literature. Many of the assumptions and constraints in the paper are arbitrary or else depend upon the technology being used. This makes any comparisons to other work difficult.

The following is a summary of some of the difficulties and deficiencies in the current PLA Folding literature.

a) The graph theoretic interpretation of the PLA folding problem is used almost exclusively. There are no alternatives available.

b) All of the present heuristic algorithms lack an analytical or empirical basis. There is no way to compare the effectiveness of the various heuristics.

c) There are a number of branch and bound algorithms for optimal PLA folding but there is no method for comparing their relative effectiveness in finding a solution.

d) Many algorithms and techniques are dependent upon arbitrary architectural constraints. There is no model for representing these constraints.

e) There is no analytical basis for expected results of PLA folding using either heuristic or optimal folding algorithms.

f) We do not know of any empirical results available on the expected foldability of PLAs.

g) There is little understanding of the nature of the problem. It is not clear as to how the fundamental characteristics of a PLA affect its difficulty in folding or even what those fundamental characteristics are.

The solutions to many of these problems are provided by the work presented in this thesis. Some of the results have been previously published elsewhere [MaT85a, MaT85b, MaT86].

Chapter 3

PLA Model

There are several different approaches used in the folding of PLAs, each of which has constraints imposed by the technology it employs. In addition, each specific technique may have several different algorithms to obtain a folding set for that type of folding. This large variety makes it difficult for one to describe a given PLA architecture or folding algorithm. There is a need for a general PLA model to represent the different types of PLAs and folding techniques.

A good model should have a number of qualities. It must represent the fundamental nature of the object it is describing without deluging the user with unnecessary details. It must be flexible enough to describe the many variants of the object yet be simple to understand. It must also be consistent with the current understanding of the problem if it is to gain acceptance in the literature.

For PLAs and PLA folding, the model must perform two basic functions. It must first accommodate a variety of implementation details and constraints. Secondly, it must concisely represent the subset of details required to describe PLA folding techniques. This representation needs to be mathematically rigorous as the model may be used in obtaining theoretical results. To satisfy the two functions, the model is divided into two parts. The first part is a topological model for representing the necessary structural information about PLAs. This allows one to describe the different architectural arrangements used in PLAs. The second part is a set-theoretic

model for representing the PLA folding problem. It is a mathematical definition of the problem to be used in describing theoretical results.

3.1. Topological Model

First a few definitions are needed,

- A *signal* is a physical representation of information. The information to be represented depends upon the logic being used and the form of the representation depends upon the technology of implementation. In the case of MOS PLAs, the values 1 and 0 are represented by high and low voltage signals.

- A *technology* is a set of physical fabrication layers and associated design rules.

- A *wire* is a line segment on a given technology layer. Associated with each wire is a signal. All parts of a wire have the same signal value at any one time. Often the terms "signal" and "wire" are used interchangeably but the term "wire" would really be describing the signal associated with it.

- Two technology layers are said to be *independent* if a wire on one layer can overlap a wire on the other with no *primary interaction* between the signals associated with each wire. Primary interaction between two signals is said to occur if the value of one signal consistently affects the value of the other.

- The *number of dimensions* of a given technology is the cardinality of the largest set of independent layers. For example, a specific NMOS technology has a metal layer, a diffusion layer, and a polysilicon layer. Transistors are formed when polysilicon crosses diffusion so these two layers are not considered independent. The largest set of independent layers has two

elements, containing either (metal and diffusion) or (metal and polysilicon). Thus this technology is said to have two dimensions.

- A *contact* is a connection between two wires that may be on different layers. The wires must overlap at the point of contact. Wires that are connected in this fashion represent the same signal value.

These definitions represent the basic building blocks for describing PLAs. No allowance is made in this model for the charging times of wires. Dynamic operations are represented as continuous discrete time steps of static events. This is not an electrical model and second order parasitic effects are not considered. The model is a topological one.

Now some functional elements can be described.

- A *switch* is a special connection between a number of wires. The switch allows one-way dependencies to exist between the signals on the wires. The nature of this dependency is determined by the definition of the switch and the technology being used. If a wire W_a affects the signals on one or more of the other wires in the connection, then W_a is called an *activating* wire. If a wire W_b is affected by the signals on one or more of the other wires then W_b is called a *dependent* wire. Power and ground wires necessary for switch operation are considered *passive* wires. It can be seen that a switch can be as simple or as complex as desired.

- A *Nor-line* is an output wire W_0, a ground wire G_0, a set of associated switches S_i , and an output signal value n. Each switch S_i, is composed of the passive ground wire G_0, the dependent wire W_0 and an activating wire A_i. A switch is said to be activated if its corresponding activating wire has a

signal value of logical one. The signal value of W_0 depends upon the switches in the following way. If any of the switches are activated then W_0 will have a signal value 0. If none of the switches are activated then W_0 will have a signal value n. The exact value of n will depend upon the technology but is usually a value of logical one. The ground wire G_0 is a passive wire necessary for proper operation of the Nor-line. It is said to be "adjacent" to the output wire and is often not shown in illustrations. Figure 3.1 illustrates a Nor-line.

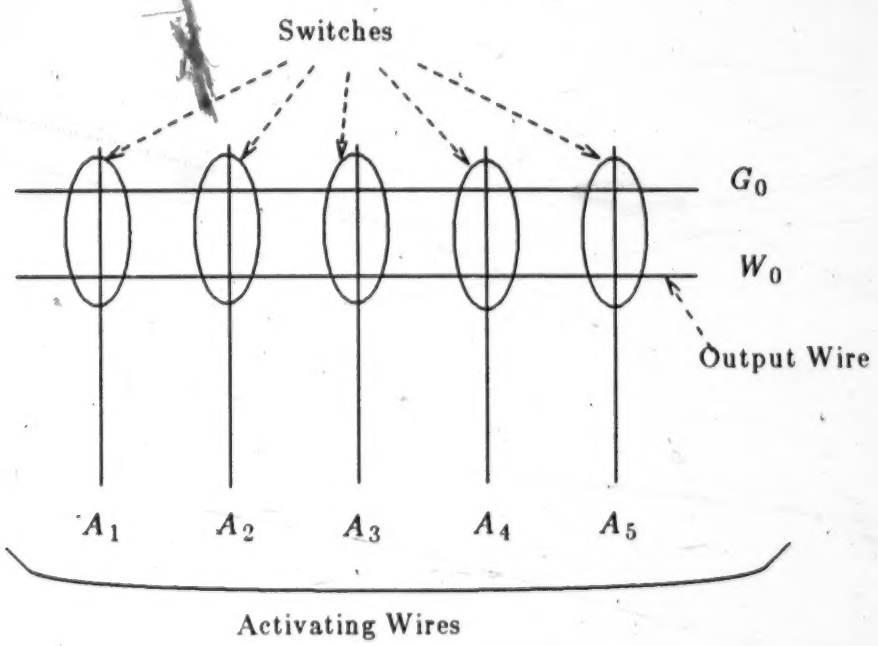

Figure 3.1 A Nor-line

- A *dimension* is a set of wires none of which intersect each other. They are usually all physically parallel. A dimension is said to be an *independent dimension* if the signals associated with the wires are all independent of each other. Note that a dimension is an arbitrary designation. Two non-intersecting wires are not necessarily in the same dimension. For

technologies which allow multiple layers of metalization, the user can specify wires on different layers to be in separate independent dimensions. A set of Nor-lines is said to be in the same dimension if their output wires are in the same dimension.

- A *simple plane* is an ordered set of Nor-lines N, all in a single dimension and a set of wires S, in another dimension each of which is an activating wire on one or more switches associated with a Nor line. An example of this is shown if Figure 3.2 where for the sake of illustration the ground wires of the Nor-lines are not drawn and switches are represented by X's .

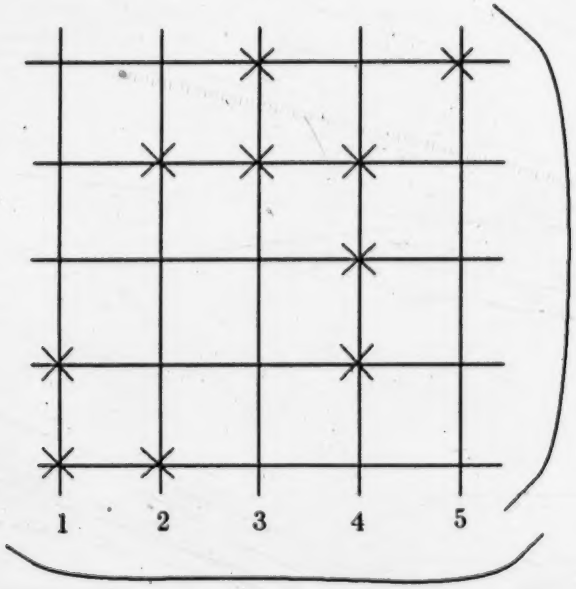

Figure 3.2 A Simple Plane

The set of wires in the set S are called the inputs to the plane and the output

wires of the Nor-lines are called the outputs of the plane. Two simple planes are said to be connected if the outputs of one plane are connected to the input of another.

These are the definitions needed to form a PLA. A PLA is composed of a set of one or more connected simple planes as illustrated in Figure 3.3. Usually there are two planes. In the Figure, the symbol "o" represent contacts between wires.

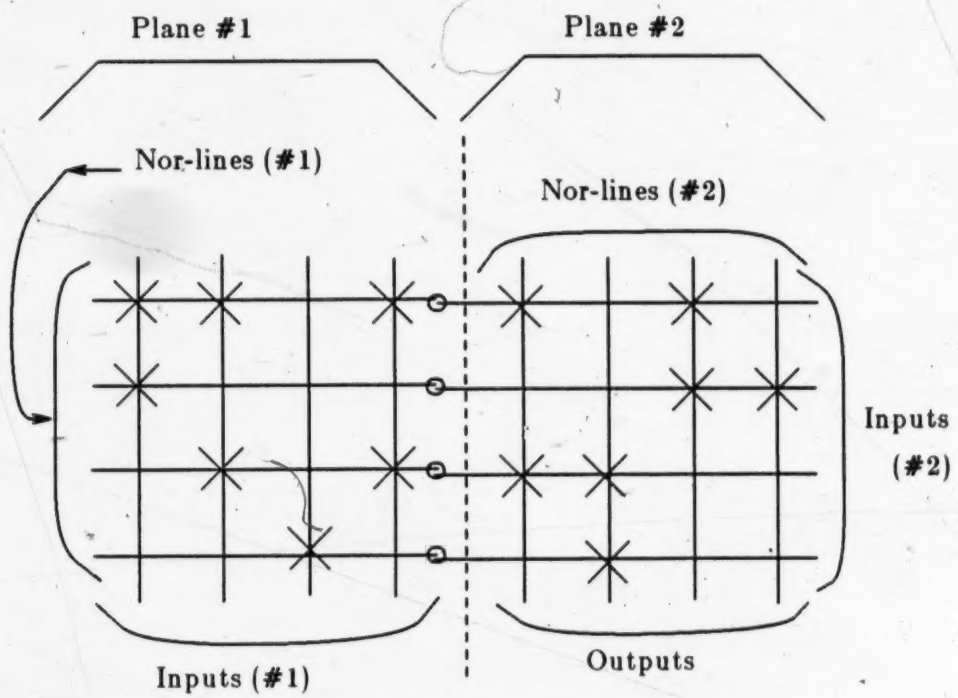

Figure 3.3 A PLA - Two Interconnected Planes

Most PLAs in use today can be described using the definitions above. For a few, however, additional definitions must be developed from the basic definitions. For example, in the description of a static CMOS PLA, a complementary simple plane would be needed. This plane would be similar

to the regular simple plane defined above except that *Nand-lines* would be used instead of Nor-lines. A Nand line produces a zero output signal when all of its input switches are activated.

This model can be used to represent PLAs when discussing folding techniques. The basis of PLA folding is that wires within a simple plane can be replaced by one or more separate wire segments. Wire segments must be collinear, as if part of a single wire, but may have different signal values. 3.4 illustrates wire segments.

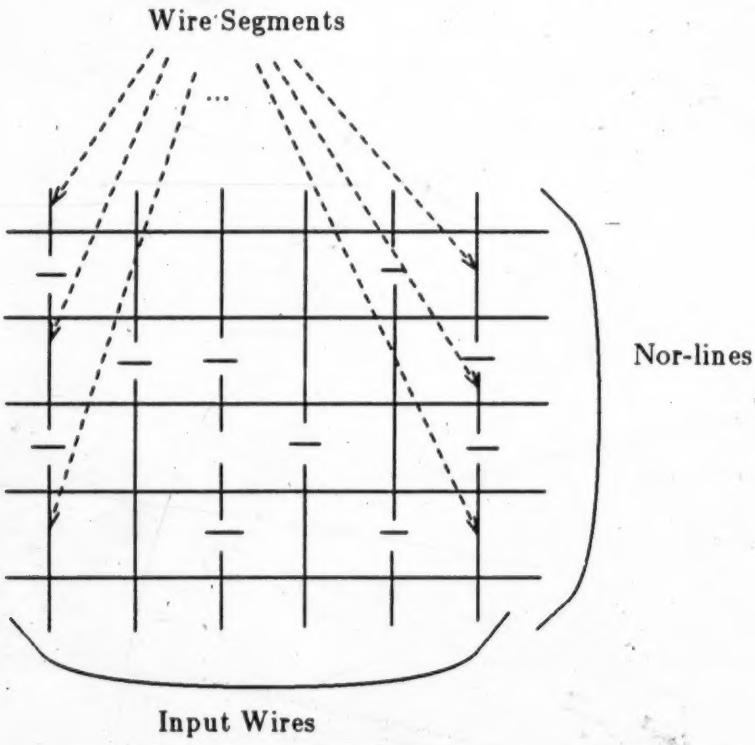

Figure 3.4 Wire Segments

The exact conditions under which wire segments are allowed is dictated by the type of folding being done. The different types of folding are described in

the next Chapter.

The above definitions complete the portion of the model used for describing topological information of a PLA. Although it has not been fully expanded here, the basic elements presented above can be used for constructing representations of most types of PLAs.

3.2. Set-Theoretic Model

For representing the nature of the PLA folding problem it is necessary to describe it in a more rigorous form than wires and planes. The following is a new set-theoretic model for this purpose. It will be shown to have certain advantages over the graph-theoretic interpretation popular in the literature.

The derivation of the model presented is based on the folding of simple planes. For ease of understanding, and consistency with current literature, the input wires are arbitrarily called columns and the Nor-lines called rows. Folding will be restricted to input wires or columns where each wire can be replaced by only two wire segments, an upper one and a lower one. An example of this type of folding is shown in Figure 3.5 and is known in the literature as simple column folding. An active intersection is a position where an input wire activates a switch which is connected to one of the Nor-lines. These are indicated by X's in Figure 3.5.

The n rows of the PLA are represented as a set R. Thus

$$R = \{r_1, \cdots, r_n\}$$

Each column of the PLA is represented by the set of rows in which it has active intersections. The m columns of the PLA are expressed as a set C of

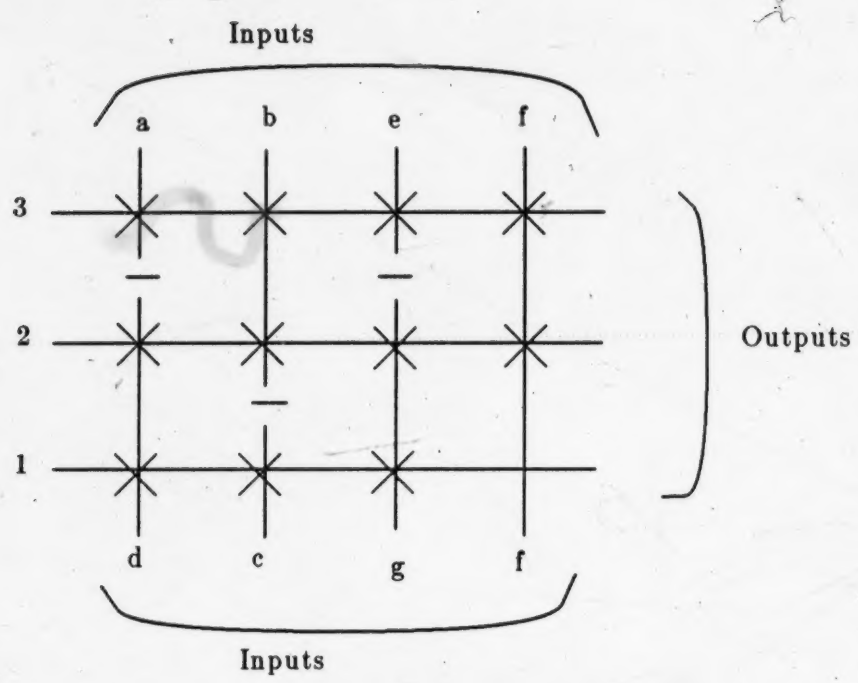

Figure 3.5 Column Folding of a Simple Plane

---

m subsets of R.

Thus

$$C=\{c_1, \cdots, c_m) \text{ where } c_i \subseteq R$$

For example, Figure 3.6 illustrates a single plane PLA. It can be represented as

$$R=\{1,2,3\}$$

$$C=\left\{a,b,c,d,e,f,g\right\}$$

$$=\left\{\{3\},\{2,3\},\{1\},\{1,2\},\{3\},\{2,3\},\{1,2\}\right\}$$

Define P to be an ***ordered partition*** of R so that

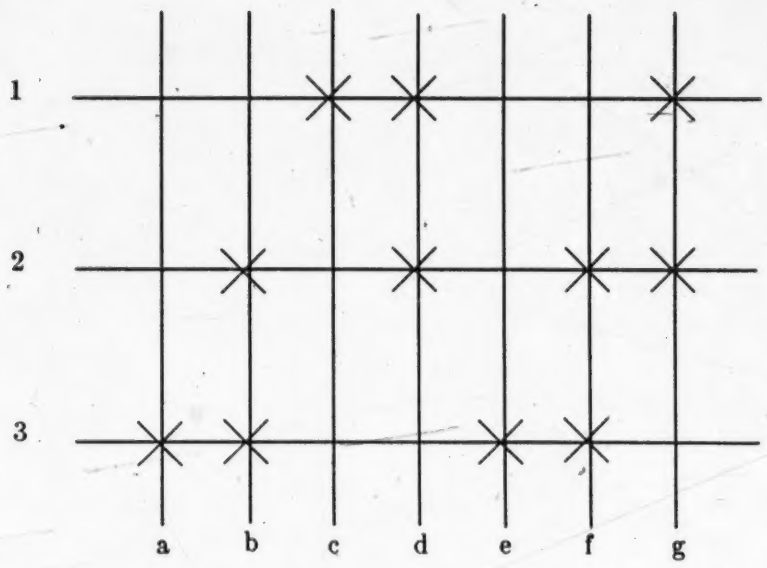

Figure 3.6 A Example Single Plane PLA

$P = (A_1, \ldots, A_t)$

where for all i,j

$A_i \subseteq R$

$A_i \neq \phi$

$A_i \bigcap A_j = \phi \quad (i \neq j)$

$\bigcup_{i=1}^{n} A_i = R$

In other words, P is an ordered set of A_i's which are non-empty, non-intersecting subsets of R.

For example, if $R = \{1,2,3,4,5,6,7\}$, an ordered partition P might be

$$P = (\{3,7\},\{2,4,6\},\{1,5\})$$

Let us define the position index rp, of a row, r_0, with respect to an ordered partition P as follows:

$$rp_P(r_0) = i \text{ such that } r_0 \in A_i$$

The position index of a given row r_0 with respect to a given ordered partition P is the position of the subset that contains the row in the ordered partition. For example, if

$$P = \left(\{3,7\},\{2,4,6\},\{1,5\}\right)$$

then $rp_P(2) = 2$ because 2 is contained in the subset $\{2,4,6\}$ which is the 2nd subset in P. Similarly $rp_P(5) = 3$ and $rp_P(7) = 1$.

Remembering that $c_0 \subseteq R$, let us define the maximum position index *maxrp*, of a c_0, (a subset of R), with respect to an ordered partion P, as follows:

$$maxrp_P(c_0) = \max\left\{rp_P(r_i) \mid r_i \in c_0\right\}$$

i.e., $maxrp_P(c_0)$ is the largest position index of all the rows in c_0. For example, if $c_0 = 2,5$ and P is defined as above, then

$$maxrp_P(c_0) = \max\left\{rp_P(2), rp_P(5)\right\}$$

$$= \max\left\{2,3\right\} = 3$$

Similarly, the minimum position index is defined as the smallest position index of all the rows in c_0.

$$minrp_P(c_0) = \min\left\{rp_P(r_i) \mid r_i \in c_o\right\}$$

where $c_0 \subseteq R$. Intuitively, the ordered partion P represents a partial ordering of the rows in R. Rows within a subset of the partition are unordered with respect to each other but are ordered with respect to rows in other subsets of the partition.

For a given column c_0, (recall that each column is represented by the set of the rows that it intersects), the maximum (and minimum) position index $maxrp_P$ ($minrp_P$) represents the relative position of the highest (or lowest) row in c_0 with respect to the partial orderings implied by P.

Let us now define a ***folding pair*** $f=(c_a,c_b)$ as an ordered pair of two elements of C.

i.e. where

$c_a \in C$

$c_b \in C$

$c_a \neq c_b$

A folding pair (c_a,c_b) is said to be ***conflict free*** with respect to an ordered partition P if

$$maxrp_P(c_a) < minrp_P(c_b) \tag{3.1}$$

That is, two columns are conflict free if they are foldable with respect to constraints implied by the ordered partition P.

A ***folding set*** $F=\{f_1, \cdots, f_n\}$ is a set of n completely distinct folding pairs where none of the folding pairs in the folding set share common elements of C.

A folding set F is said to be ***conflict free*** if

There exists an ordered partition P such that all members of F are conflict free with respect to to said P.

A folding set is conflict free if an ordering of the rows can be found that allows all the folding pairs in the folding set to be conflict free. Thus the PLA folding problem can be defined as follows.

Given the sets R and C, find the conflict free folding set F of maximum cardinality.

Note that this will, of course, also include finding the ordered partition P which corresponds to the folding set F. The partition is just a representation of the partial ordering of the rows that does not conflict with any row orders imposed by folding pairs in the folding set. This is illustrated in the folded PLA in Figure 3.7 where for the sake of illustration, the folding pairs have been ordered by the wire cut positions.

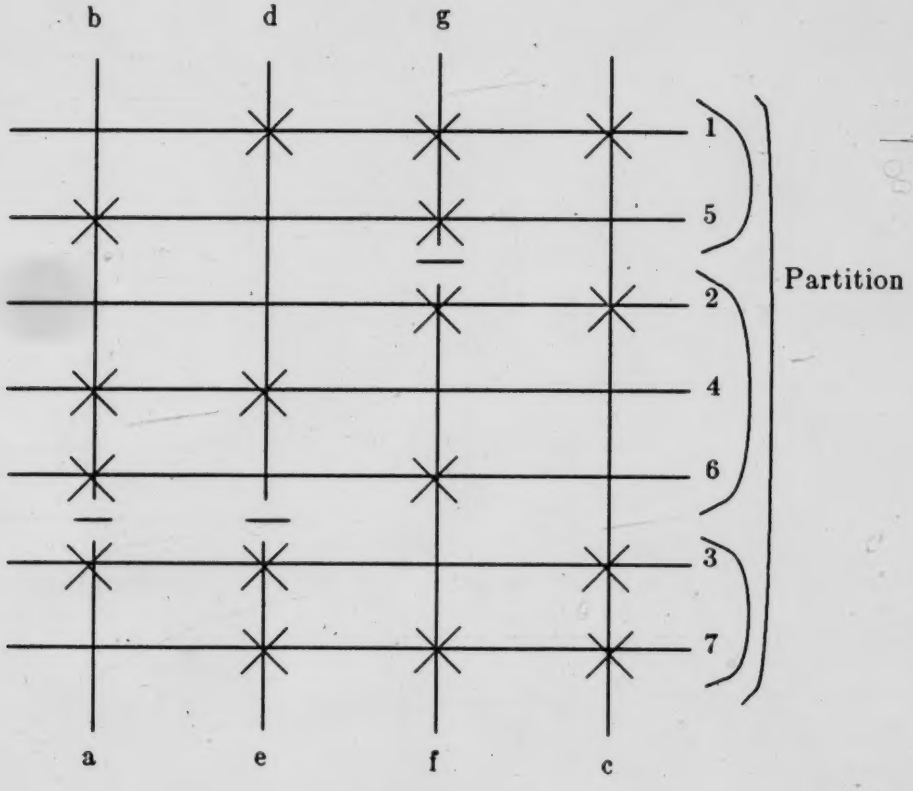

Figure 3.7 Ordered Partition of a PLA

The exact arrangement of the partition is dictated by the positions of the

wire cuts in each of the folding pairs.

There are a number of advantages that the above set theoretic definition of the PLA folding problem has when compared to the graph theoretic interpretation currently used in the literature [HSN80] . First it is reasonably easy to understand. Basically, two columns are foldable if the lowest row in the upper column is above the highest row in the lower column (see Equation 3.1). Secondly, this model captures the essence of the problem without presenting any details of the algorithm implementation. While transitive closure of the row orders may be necessary in the implementation of certain algorithms dealing with PLAs, it is not necessary in a mathematical model whose purpose is to aid understanding of the problem. Finally, as an alternative model to the one which has been almost exclusively used in the literature, it should generate new ideas and methods for folding PLAs.

Chapter 4

PLA Folding Taxonomy

The recent literature on PLA folding contains a variety of different techniques and algorithms. Comparison is difficult because authors include constraints which are particular to their type of folding. This Chapter presents a PLA folding taxonomy for classifying the different folding methods. The taxonomy is divided into three separate parts. First, a classification scheme for the architectural structure of PLAs is presented. This scheme is based on many of the elements described in the topological model of the previous Chapter. Second, a terminology is presented for classifying the various folding techniques. This classification is based on fundamental characteristics common to all techniques. Finally, a categorization of the different PLA folding algorithms is provided. The algorithms are distinguished by their fundamental properties rather than by the type of folding which they perform. There are common elements among algorithms even though they may perform different types of PLA folding.

There is already a taxonomy available in the literature [HNS82a], but it is limited to a simple terminology for describing folding techniques which are currently available. The taxonomy presented here is more complete and better represents the nature of PLA folding. It should prove to be an effective classification scheme for new types of folding and yield to a better understanding of the PLA folding problem.

4.1. Architectural Classification Scheme

This first section describes a method for classifying PLAs by architectural considerations. This method distinguishes PLAs by examining different fundamental features as well as the implementation technology. Many of the features are those which are described by the topological model in the previous Chapter. They are as follows.

1. Number of Layers

This is a function of the technology used to implement a PLA. As an example, standard NMOS technology has three layers, polysilicon, diffusion and one layer of metal. Many of the modern VLSI technologies now include two layers of polysilicon and two or more layers of metal. Each layer is not necessarily independent of all other layers.

2. Number of Dimensions

The number of dimensions of a technology is defined in the PLA model of the previous Chapter as the size of the largest set of independent layers. The same NMOS technology mentioned above would be considered two dimensional. The number of dimensions is an important classification criteria. Most PLAs today are rectangular in shape. They are composed of two sets of wires, one set running horizontally, and one vertically. This is a result of most VLSI technologies having two dimensions, that is at most two independent layers. Now that modern technologies contain additional layers of polysilicon and metalization, PLAs are no longer restricted to the rectangular layout. There can now be PLAs composed of three or four sets of wires, each set being aligned in a different direction. Although the exact

layout of such a PLA is beyond the scope of this work, it is clear that the number of dimensions is an important measurement concerning the limitations of the technology.

The above two classifications pertain directly to the technology used to implement the PLA. They are sufficient to express the fundamental limitations of a specific technology. However, they do not describe such details as exactly how specific layers interact with other layers and which layers transmit signals faster than others. If such details are needed, then a full description of the technology must be provided.

3. Number of Levels

The number of levels of a PLA is defined as the number of planes in the longest path through a set of interconnected simple planes. It represents the number of levels of logic that can be implemented by the PLA. Most PLAs today are two level and implement simple sum-of-products And-Or logic equations. However some PLAs include decoding logic on the inputs to the PLA and this effectively gives the PLA three levels.

4. Timing Considerations

PLAs may be either dynamic or static. Static PLAs require no timing signals to produce a valid logical output. However, they are usually designed as part of some other synchronous circuit. Dynamic PLAs, on the other hand, require constant timing pulses to charge wires and produce correct results. Typically static PLAs take up more chip area but require less

complex external circuitry. Both types are widely used today.

5. Types of Simple Planes

PLAs can be classified by the types of simple planes they are composed of. Recall that simple planes are sets of wires as defined in the previous Chapter. PLAs that contain only one type of simple plane are termed *simple PLAs*. Those that contain two different types of planes are termed *complementary PLAs*. Static CMOS PLAs are complementary requiring both a simple plane and a complementary plane to function. Most other PLAs are simple PLAs.

Note that there is no distinction between the AND-plane and the OR-plane. These are classifications created by authors for specific PLAs. This is not to say that they are incorrect terms. They are just not part of this taxonomy as they become meaningless terms for more complex PLAs.

This concludes the classification of the architectural structures and technology of PLAs.

4.2. Classification of Folding Techniques

There is even more variety in the types and forms of folding techniques than in structures of PLAs. Although there are variations specific to each PLA type, there are nonetheless a number of classification criteria which are fundamental to all types.

6. Segmentation Number

PLA folding is the replacement of individual wires within a PLA by one or more wire segments. Often there is a limit to the number of wire segments

allowed to replace a single wire. This limit is known as the segmentation number. If the segmentation number is two, the folding is termed "simple folding". If the segmentation number is three or more, then the folding is termed "multiple folding". These definitions are consistent with the terminology currently used in the literature.

7. Number of Folded Dimensions

Folding techniques can be distinguished by the number of separate dimensions in which folding occurs. If folding is restricted to only one dimension, the PLA is said to be "singly folded". "Doubly folded" describes PLAs where folding occurs in wires contained in two dimensions, and so on. As most PLAs contain at most two dimensions, there are hardly any examples of PLAs with three or more folded dimensions.

8. Open and Closed Ended Folding

Wires within a PLA can be classified according to various constraints. Each wire must have at least one end connected to another wire or external connection. This follows from the idea that there must be at least one way, topologically, to get a signal to or from the wire, otherwise the wire would serve no purpose. If this connection is the only one necessary for the proper functioning of the wire, then the wire is said to be "open ended". This definition is derived from the fact that at least one end of the wire is open or free to be used by the PLA folding technique. It is not required for the wire to function. If, in addition to the input/output connection to the wire, another connection such as a Nor-line driver is needed for proper functioning of the wire, the wire is said to be "closed ended". Both ends of the wire are

used in its normal operation and are unavailable for other uses.

Input wires in a simple plane are open ended. The only external connection to the wire is the input signal. This leaves one end of the wire free for use in folding. Nor-lines, on the other hand, are closed ended wires. One end of the Nor-line is used for the output signal that it generates and the other end is connected to some sort of driving device. The exact nature of the driving device depends upon the technology being used.

Note that connections to switches internal to a simple PLA plane are not considered significant in determining whether a wire is open or closed ended. This distinction is a topological one and depends only on connections made at the end of the wire.

Using the above definitions, PLA folding can be classified by whether folding occurs on open or closed ended wires. It is clear that open ended folding is much simpler than closed ended folding. Figure 4.1 illustrates open ended folding on a simple plane. The inputs to the plane run vertically and are folded. The Nor-lines run horizontally and are not folded. The drivers to the Nor-lines are represented by the electrical resistor symbol. Figure 4.2 illustrates closed ended folding. Note that without an extra dimension such as a second layer or metal, it is impossible to retrieve the output signals from the Nor-lines. This type of folding has not been used much to date because of this constraint.

9. Internal and External Folding

Each simple plane has a set of wires which are the inputs to that plane. If those inputs are connected directly to signals external to the PLA, they are

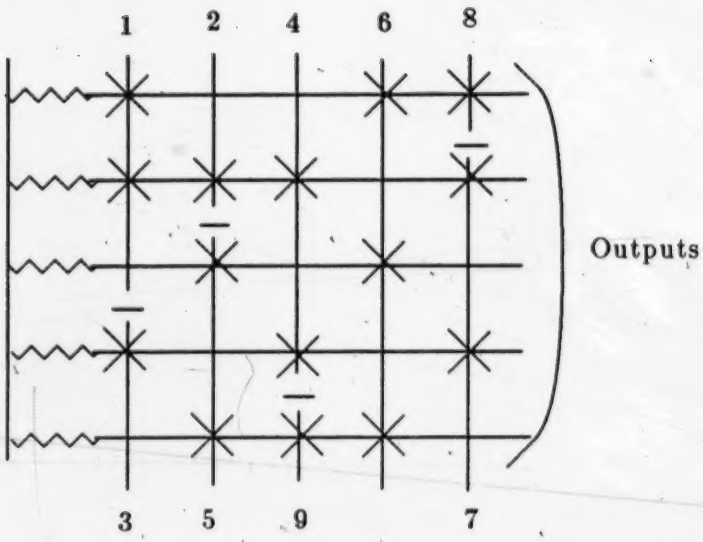

Figure 4.1 Folding of Open Ended Wires

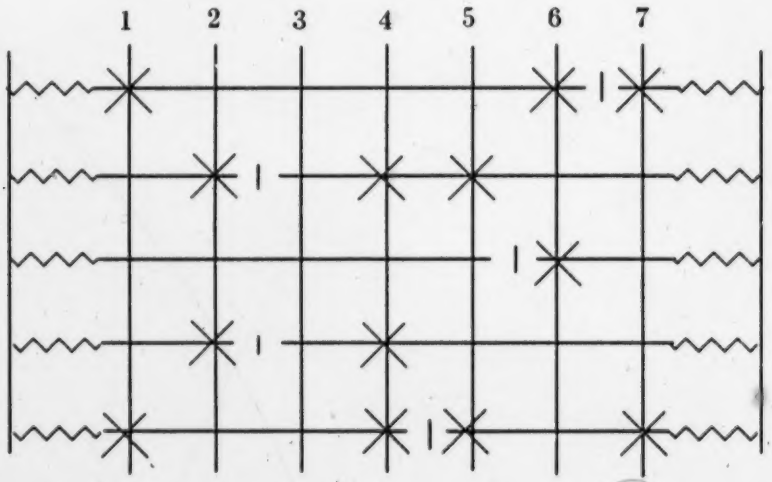

Figure 4.2 Folding Closed Ended Wires

termed "primary inputs" to the PLA. If the input signals are generated from some point within the PLA, such as from another simple plane inside the PLA, then they are termed "local inputs".

"Internal folding" is the folding of wires that are not primary inputs. This includes the folding of all Nor-lines and of all wires that are connected directly to Nor-lines within the PLA. "External folding" is the folding of the primary inputs to the PLA. Generally internal folding is more difficult than external folding. Figure 4.3 illustrates a two plane PLA with external folding.

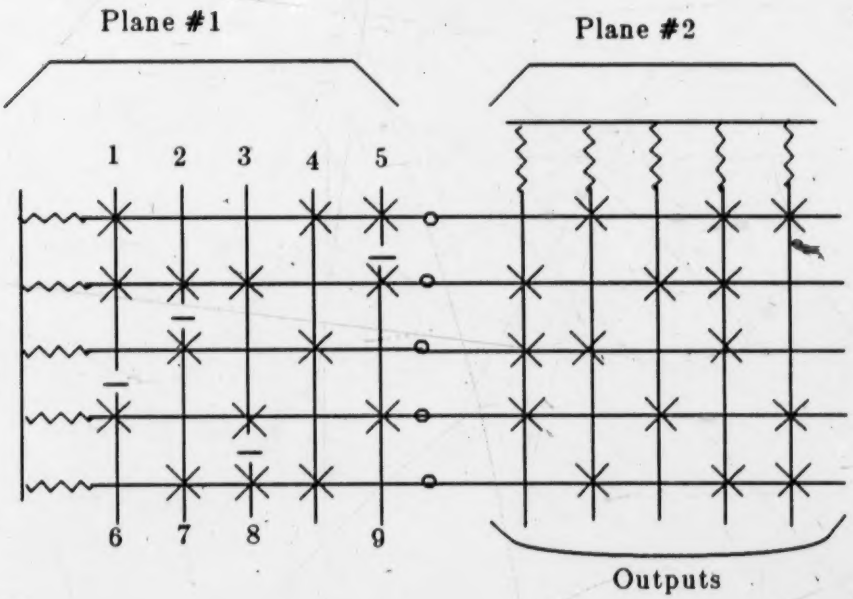

Figure 4.3 External Folding of a Two Plane PLA

In comparison, Figure 4.4 illustrates a three plane PLA with internal folding. The outputs of plane #1 and plane #3 are folded together as the local inputs to plane #2. This particular example is known in the literature as the AND-OR-AND arrangement as it can be viewed as an OR plane between two AND planes. However, this is an arbitrary categorization as the vertical Nor-lines in Figure 4.4 could be intermixed with the vertical inputs. Remember that a simple plane consists of a set of Nor-lines. There is no stipulation that the

lines must all be adjacent in a regular shape.

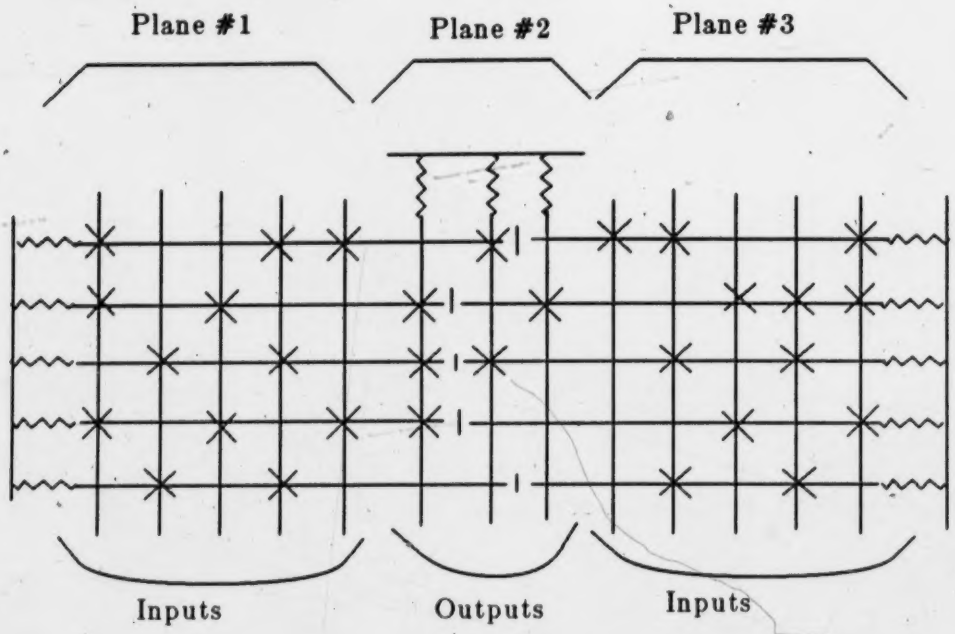

Figure 4.4 Internal Folding of a Three Plane PLA

---

10. Separate and Mixed Folding

It can be seen that a given PLA, with multiple planes, can contain an assortment of different folding techniques. If, for each simple plane, only one dimension is allowed to be folded then the PLA is said to contain only "separate folding". If, however, there is at least one simple plane where folding is allowed in two or more dimensions (i.e. row and column folding within a single plane) then the PLA is said to have "mixed folding". Algorithms for dealing with separate folding are simpler than those for dealing with mixed folding.

11 & 12. Input and Output Access

As a PLA is not a complete device in itself, it must have inputs and outputs accessing the outside world. If all of the primary inputs to the PLA are in a single dimension, the PLA is said to have "one sided input access". If they are contained in only two different dimensions, the PLA is said to have "two sided input access" and so on. For three or more sided access the term "multi sided input access" is sufficient. If some of the inputs are not topographically accessible on the periphery of the PLA, the inputs are said to be "internally accessible". This occurs, for example, in multiple folding. The same definitions apply to accessing the outputs of the PLA.

13 & 14. Internal and External Constraints

There are a number of additional constraints which can occur within a PLA. Suppose, for example, the simple planes were required to be arranged in the AND-OR-AND fashion seen in Figure 4.4. This and other such architectural constraints are termed "internal constraints".

There can also be constraints imposed on the primary inputs to the PLA. These are termed "external constraints" and usually consist of orderings imposed upon the inputs. If the ordering imposed is complete, the inputs are said to be "fully ordered". If the ordering is incomplete, the inputs are said to be "partially ordered". If the ordering is in the form of limits on the relative position of certain inputs, the inputs are said to be "bounding ordered". Finally, if a set of orderings is not mandatory but instead is preferred, then the ordering is said to be a "weighted cost ordering". This would occur where the position of two inputs could be interchanged by, for example, changing from a metal to a polysilicon layer. The designer would assign an area cost to changing layers and would need to determine if the

benefits gained by the new arrangement in the PLA were worth using it.

The above criteria are sufficient for describing all of the folding techniques in use today as well as foreseeable future ones. As new multiple metal and polysilicon layer technologies emerge in VLSI, there will be a tendency to the use of more complex PLAs. The main advantage of the above definitions is their adaptability to these new multiple-level, multiple plane PLAs.

4.3. Categorizing Folding Algorithms

Each folding algorithm is designed to implement a particular kind of PLA folding. However, aside from this, there are a number of elements common to all algorithms, which allows us to categorize them.

15. Optimality

The object of PLA folding is to reduce the area of the PLA based on some area measurement criteria. Algorithms can be classified by whether they find the theoretically optimal area reduction or not. Those that find an optimal solution are termed "optimal algorithms" and those that do not are termed "non-optimal" algorithms. The non-optimal algorithms are usually heuristic. The optimal algorithms usually use some sort of branch and bound method. A later Chapter in this thesis will describe methods for comparing the performance of different optimal and non-optimal algorithms.

16. Search Tree Backtracking

In any heuristic directed search for near optimal folding sets, the algorithm will select folding pairs to be included in the final folding set. If these selections are final, and no backtracking occurs, the algorithm is said to

be a "direct search". If the algorithm can adapt to poor choices and change selections it is said to be an "adaptive search".

17. Deterministic and Random Types

If the heuristic used to select column pairs during the folding process is based totally on specific properties of the PLA , it is said to be "deterministic". If the heuristic selection process has a random element to it and does not necessarily produce the same results each time, it is said to be a "random type" algorithm. The extreme of this concept is the random selection heuristic presented later in this thesis which is based totally on random column pair selection. This is shown to be more effective than the deterministic algorithms in use today.

4.4. Summary and Application

Table 4.1 shows a concise summary of the taxonomy. Table 4.2 shows an analysis of some of the representative PLA folding systems and results in the literature, using this taxonomy. The selections are specified by their reference number as found in the References section of this thesis. Note that in the table na= not applicable and ns= not specified. It can be seen that most current folding is fairly simple as compared to what can be represented in the taxonomy.

PLA Classification

1. Number of layers
2. Number of dimensions in the technology
3. Number of levels
4. Timing considerations (static/dynamic)
5. Type of planes (simple/complementary)

PLA Folding Technique Classification

6. Segmentation number (simple/multiple)
7. Number of folded dimensions
8. Open/Closed ended folding
9. Internal/External folding
10. Separate/Mixed folding
11. Input access (one/two/multi sided, internal)
12. Output access (one/two/multi sided, internal)
13. Internal constraints
14. External constraints (fully/partial/bounded/weighted cost orderings)

Categorization of Folding Algorithms

15. Optimality
16. Backtracking (direct/adaptive)
17. Deterministic/Random Type

Table 4.1 Summary of Taxonomy

Taxonomy Category	Literature Reference			
	[Pai81]	[EgL84]	[Gra82]	[MiS83b]
1. # Layers	3 / 4	ns	ns	3 / 4
2. # Dimensions	2 / 3	3	2	2 / 3
3. # Levels	1	2	ns	2
4. Timing	ns	ns	ns	ns
5. Plane Type	simple	simple	simple	simple
6. Segmentation #	multi	2	2	multi
7. # Folded Dims	1	1	1 / 2	2
8. Open/closed	open	open	open	open
9. Intern/Extern.	int/ext	ext.	ext.	int/ext
10. Separate/Mixed	sep.	sep.	sep/mixed	mixed
11. Input Access	1,int.	1	1	1
12. Output Access	1	1	1	1
13. Intern Constr.	none	bipart.	none	yes
14. Extern Constr.	none	none	none	order
15. Optimality	non-opt	non-opt	opt	non-opt
16. Dir/Adapt.	direct	adapt	adapt	direct
17. Rand/Determ.	determ	determ	determ	determ

Table 4.2 Application of Taxonomy to Current Literature

Chapter 5

Optimal PLA Folding

This Chapter presents some results on the optimal PLA folding problem. Recall that the optimal folding of a PLA is the folding arrangement which yields the theoretically greatest chip area savings within the constraints of the type of folding being used. The problem of finding this folding arrangement is known as the optimal PLA folding problem and has been shown to be np-complete [HSN80]. This Chapter confines itself to the study of simple column folding as was defined earlier and illustrated in Figure 2.2. First an analysis is performed of the number of distinct folding sets possible for a given PLA.

A *Basic Optimal Search Algorithm* is presented and used in the implementation of a new Branch and Bound algorithm for optimal PLA folding. This new algorithm is used to generate some empirical data as to how the foldability of random PLAs is affected by some of its fundamental properties. As well, two new *bounding measures* are introduced. The first one represents the inherent difficulty of finding an optimal folding set for a given class of PLAs and the second one represents the relative effectiveness of a given algorithm in solving this problem. Both measures are shown to be useful in comparing different bounding techniques.

5.1. Size of the Problem

Recall from Chapter 3 that the PLA folding problem can be defined as follows.

> Given the sets R and C (representing the rows and columns respectively), find the conflict free folding set F of maximum cardinality.

As a first step towards producing a solution to this problem, one may first count the number of possible distinct folding sets, conflict free and otherwise, that occur. First a definition is required.

> Two folding sets F_1 and F_2 are said to be *non-distinct* if
>
> 1) one is a complete inversion of all of the folding pairs of the other or
>
> 2) one is a rearrangement of the folding pairs of the other or
>
> 3) one is a transformation of the other through a combination of 1 and 2.

If none of the above is true then the two folding sets F_1 and F_2 are said to be *distinct*.

For example, in a PLA with 8 columns, if

$$F_1 = \{(1,6),(2,4),(3,8),(7,5)\},$$

and

$$F_2 = \{(6,1),(4,2),(8,3),(5,7)\},$$

then F_1 and F_2 are *non-distinct* because one is a complete inversion of the other.

Another example; suppose

$$F_1 = \{(1,6),(2,4),(3,8),(7,5)\},$$

and

$$F_2 = \{(2,4),(7,5),(1,6),(3,8)\},$$

then F_1 and F_2 are again *non-distinct* because one is simply a rearrangement of the folding pairs of the other. As a final example, note that the following are all distinct folding sets,

$$\{(1,6),(2,4),(3,8),(7,5)\},$$
$$\{(1,2),(4,8),(7,3),(5,6)\},$$
$$\{(4,2),(8,5),(7,6),(3,1)\}$$

and the following are all non-distinct folding sets.

$$\{(1,2),(3,6),(7,8),(5,4)\},$$
$$\{(2,1),(6,3),(8,7),(4,5)\},$$
$$\{(2,1),(8,7),(4,5),(6,3)\}$$

Note that the total number of possible distinct folding sets depends only on C, the number of columns of the PLA. It is assumed for this analysis that C is even. Similar results occur if C is odd.

A function $G_C(n)$ is defined as follows.

$G_C(n)$ = the number of distinct folding sets that contain *exactly* n folding pairs for PLAs of C columns.

First,

$$G_C(1) = \frac{C \times (C-1)}{2}$$

The division by two is necessary because folding sets such as

$$\left\{(C_a, C_b)\right\} \text{ and } \left\{(C_b, C_a)\right\}$$

are not considered distinct. Now,

$$G_C(2) = \frac{C \times (C-1) \times (C-2) \times (C-3)}{2 \times 2!}$$

The division by 2! is necessary because the folding pairs in the folding set can be arranged in 2! different ways. For example, foldings sets such as

$$\left\{(C_a, C_b), (C_d, C_e)\right\} \text{ and } \left\{(C_d, C_e), (C_a, C_b)\right\}$$

are not considered distinct. In general

$$G_C(n) = \frac{\prod_{i=0}^{2 \times n-1} (C-i)}{2 \times n!}$$

or by the recurrence relation

$$G_C(n) = G_C(n-1) \times \frac{(C-2 \times n+2) \times (C-2 \times n+1)}{n}$$

$T(C)$ is number of distinct folding sets of all sizes for PLAs of C columns. It can be defined as follows.

$$T(C) = \sum_{i=0}^{\frac{C}{2}} G_C(i)$$

$T(C)$ represents the total number of distinct folding sets for PLAs of C columns. Table 5.1 shows the value of $T(C)$ for various values of C.

C	2^C	$T(C)$	$C!$
2	4	1	2
4	16	12	24
8	256	2968	40320
12	4.10e+03	1.80e+06	4.79e+08
20	1.05e+06	3.65e+12	2.43e+18
30	1.07e+09	2.23e+21	2.65e+32
40	1.10e+12	6.67e+30	8.15e+47

Table 5.1 Counting distinct folding sets

It can be seen that $T(C)$ grows very quickly with respect to C and appears to be of an order greater than exponential but less than factorial.

The fact that $T(C)$ appears to grow at least exponentially concurs with the prior result that the problem is "np-hard". If $T(C)$ grew polynominally then a solution could be found in polynominal time by simply examining all of the $T(C)$ distinct folding sets.

5.2. New Branch and Bound Algorithm

The basis of any algorithm to find the optimal folding set of a PLA is to search all of the distinct folding sets to find the largest set which is conflict free. One wants to ensure that non-distinct folding sets are examined only once. In fact, not all distinct folding sets need be examined. As an example, if the folding set

$$\{(1,3),(2,5)\}$$

is found not to be conflict free and thus unimplementable, then all folding sets containing these two folding pairs would also be unimplementable. For example,

$$\{(1,3),(8,6),(2,5)\}$$

would not be conflict free and would not need to be examined. (It contains both the folding pairs of the previous folding set.) This condition can be described by the following theorem. The term *conflict free* is as defined in the set-theoretic model of chapter 3.

Theorem 5.1

If G and F are folding sets for a PLA where

G is not conflict free and

$G \subset F$

then

F is not conflict free.

Proof

The proof is by contradiction. Suppose F were conflict free. Given the definition of conflict free as found in Chapter 3, this would mean

There exists an ordered partition P such that all members of F are conflict free with respect to said P.

Since $G \subset F$, all members of G are also conflict free with respect to the same P as well so G is conflict free. This contradicts the original statement that G

is not conflict free.

Thus F must be not conflict free. *Q.E.D.*

The search tree that will be used is defined in a manner similar to that in [LeL84], where a lexicographical order is imposed upon all of the folding sets and they are examined in order. The lexicographical order used here is as follows. The columns of the PLA are arbitrarily numbered from 1 to n. All of the possible folding pairs are ordered first by the minimum column number in the pair and then by the maximum column number in the pair. For example (2,5) < (4,3) and (4,6) < (7,4). For column pairs that are equal by this rule, the pair where the minimum column occurs first is ordered first i.e. (1,2) < (2,1). For each folding set F, the folding pairs in the set are arranged in order from highest to lowest and the folding sets are ordered lexicographically with respect to the order of the contained folding pairs. e.g.

$$\{(9,7),(5,6),(1,2)\} < \{(9,4),(4,3),(2,1)\}$$

and

$$\{(9,7),(5,6),(1,2)\} < \{(9,7),(6,5),(1,2)\}$$

Note that this ordering is completely arbitrary.

This ordering of the folding sets is used to construct a *Basic Optimal Search Algorithm* for finding the optimal folding set of a PLA. The following functions are used in the algorithm and are briefly described. All the functions have been completely implemented. Some of the functions are recursive and are invoked at different points in the search tree. Each function has access to a global set of constraints imposed by the current set

of folding pairs. These global constraints are updated as the procedure moves up and down the search tree.

Findfolds(a,b) is the main recursive procedure which takes as an argument the last column pair examined. Each call to this routine at any level represents the examination of a distinct folding set. At the main level it is called to search all folding sets which it does by recursively calling itself.

Foldable(i,j) is a logic procedure which determines whether the column pair (i,j) is conflict free with respect to the global set of constraints imposed by prior column pairs.

Fold(i,j) is a procedure which adds any constraints imposed by folding the column pair (i,j) to the global set of constraints.

Unfold(i,j) is a procedure which removes the constraints which resulted from folding column pair (i,j) from the global set of constraints.

ncols is the number of columns of the PLA.

The framework of the algorithm is as follows.

```
Findfolds(a,b) {

  for i= 1+min(a,b) to ncols do {
    for j= i+1 to ncols do {
      if (Foldable(i,j)) then do {
        Fold(i,j)
        Findfolds(i,j)
        Unfold(i,j)
      }
      if (Foldable(j,i)) then do {
        Fold(j,i)
        Findfolds(j,i)
        Unfold(j,i)
} } } }
```

Findfolds is originally called from the main level with the first column pair *Findfolds(1,2)*. It recursively calls itself to search all of the folding sets in

lexicographical order. This ordering ensures that each distinct folding set is examined only once and that any folding sets eliminated by reason of Theorem 5.1 are not examined.

Remember that folding pairs are ordered first by the minimum column in the pair and then by the maximum column in the pair. This is why the outer For loop begins at one plus the minimum of the previous column pair selected. For example, if the last column pair selected was (5,9) then the outer For loop would need only begin at column 6. Any column pairs lexicographically less than (5,9) have already been examined and any column pairs lexicographically between (5,9) and (6,*) will contain the column 5 which has just been folded and is thus unavailable.

The inner For loop begins at one plus the current value being used in the outer For loop. The reason for this is as follows. Suppose the outer For loop is currently at column 6 and the inner For loop is at column 9. The two If statements will try the two column pairs (6,9) and (9,6) respectively. Now, at a later time the outer loop is at column 9. There is no need for the inner loop to examine column 6, for instance, because the column pair (9,6) has already been examined when the outer loop was at column 6. So the inner For loop need only examine columns greater than the column in the outer loop.

A bounding procedure is used for further limiting the number of folding sets that must be examined without sacrificing optimal folding. A global variable *best-so-far* is created which represents the size of the largest folding set found so far. A function *Limit()* is created which can be called at any level. This function is based on some theoretically based method and returns

an upper limit on the size of the folding set achievable from the current location in the search tree and the global set set of folding constraints. The basic optimal search algorithm can now be defined as follows:

```
Findfolds(a,b) {
  best-so-far = max(current folding set size,best-so-far)

  if ( best-so-far >= Limit() ) then {
    return
  }

  for i= 1+min(a,b) to ncols do {
    for j= i+1 to ncols do {
      if (Foldable(i,j)) then do {
        Fold(i,j)
        Findfolds(i,j)
        Unfold(i,j)
      }
      if (Foldable(j,i)) then do {
        Fold(j,i)
        Findfolds(j,i)
        Unfold(j,i)
} } } }
```

One can see that the *Limit()* function has the potential of pruning large portions of the search tree.

The above algorithm is called the *Basic Optimal Folding Algorithm*. It can be used to implement various branch and bound methods for optimal PLA folding. The only change between methods is the exact technique used in the *Limit()* function to produce an upper bound on the size of the folding set. There are a number of good techniques in the literature [Gra82, LeL84].

The new bounding method , or *Limit*() function presented here is as follows. A global list is maintained of which columns have already been folded for the current folding set and which ones remain to be folded. At each stage in the search tree two columns from the set of K unfolded columns are selected and marked as unavailable. These two columns are arbitrarily

chosen. The Findfolds() function then calls itself to attempt to increase the current folding set using only the remaining $K-2$ columns. If this attempt results in h new folds being obtained, then the upper limit on the number of new folds obtainable at that stage in the search space is $h+2$. In other words, if the two selected columns are returned to available status, the best that could result is to find two additional folds. This is the *Limit*() function. These two additional folds would be possible new folding pairs containing each of the previously unavailable columns.

```
Limit() {
  Mark two of the remaining columns as unavailable;
  Call Findfolds(a,b);
  h= # of folds found by Findfolds();
  Restore the two unavailable columns;
  return(h+2);
  }
```

As an example, suppose the available columns are

(7,8,9,10,11,12,13,14,15,20,22,23,25,27,28)

Columns 7 and 8 then are marked as unavailable and the remaining columns are folded to attempt to increase the size of the current folding set. Suppose 3 new folds result. If column 7 and column 8 are now marked available, the best one could expect is 5 new folds. This would be at least the 3 previous folds, plus possibly one new folding pair containing column 7, plus possibly one new folding pair containing column 8. It is clear that 6 or more new folds would not be possible for at least 4 of them would have to contain neither column 7 nor column 8 and thus those 4 would have been found when column 7 and column 8 where marked unavailable. If the upper bound of 5 new folds from this point in the search tree is not enough to beat a previous folding set then the algorithm does not have to perform any additional searching of this

branch of the tree.

Of course determining the upper bound is not cost free. The basis of the method is that the cost of searching the number of nodes used in bounding is more than made up for by nodes pruned from the tree. This is as expected as the bounding subset of the search tree generally is smaller than the subset which needs to be searched without bounding. Later in this Chapter a bounding measure will be described for quantitatively measuring how well this bounding methods works and comparing it with other bounding techniques.

5.3. Empirical Data

The above Branch and Bound algorithm for optimal PLA folding has been fully implemented in the C language on a Vax 11/780. This algorithm was used in a study to experimentally observe the effects of fundamental properties on the foldability of random PLAs. These PLAs all had densities which were evenly distributed across the columns and were generated using the algorithm described in the appendix. First the effects of the density of the PLA were examined. Figure 5.1 illustrates the results. Each single data point in the graph represents the size of the optimal folding set for PLAs of size r = 20 and c = 18 and varying density as averaged over 300 optimally folded PLAs. The graph appears to indicate that *complete folding* of the PLA occurs up to some critical density, after which the optimal folding set size drops sharply. Complete folding is defined to be when all of the columns of the PLA fold. For further increases in density, the foldability continues to drop, but slowly tapers off until a density of 50%, where no folding occurs.

The point at which complete folding ends and the sharp drop occurs is termed the *Complete Folding Density Limit* or *CFDL*. This *CFDL* will be different for PLAs of different sizes. Optimal folding up to the *CFDL* usually requires comparatively little computer time to perform.

The second part of the study deals with the effect of the number of columns on the foldability of a PLA. This is quite important as it is the number of columns which most dramatically effects the computer time required to perform Optimal PLA Folding. Figure 5.2 shows a plot of the size of the Optimal Folding Set versus the number of columns in a PLA. The number of rows and density for all of the PLAs were kept constant at 16 and 25% respectively. Again, each single data point on the graph was derived by averaging the results over 300 optimally folded PLAs. The graph indicates that the foldability of a PLA appears to increase linearly with respect to the number of columns. This observation is useful in predicting the foldability of PLAs with c greater than $\approx$ 30, where optimal folding can become prohibitively expensive, even using branch and bound techniques.

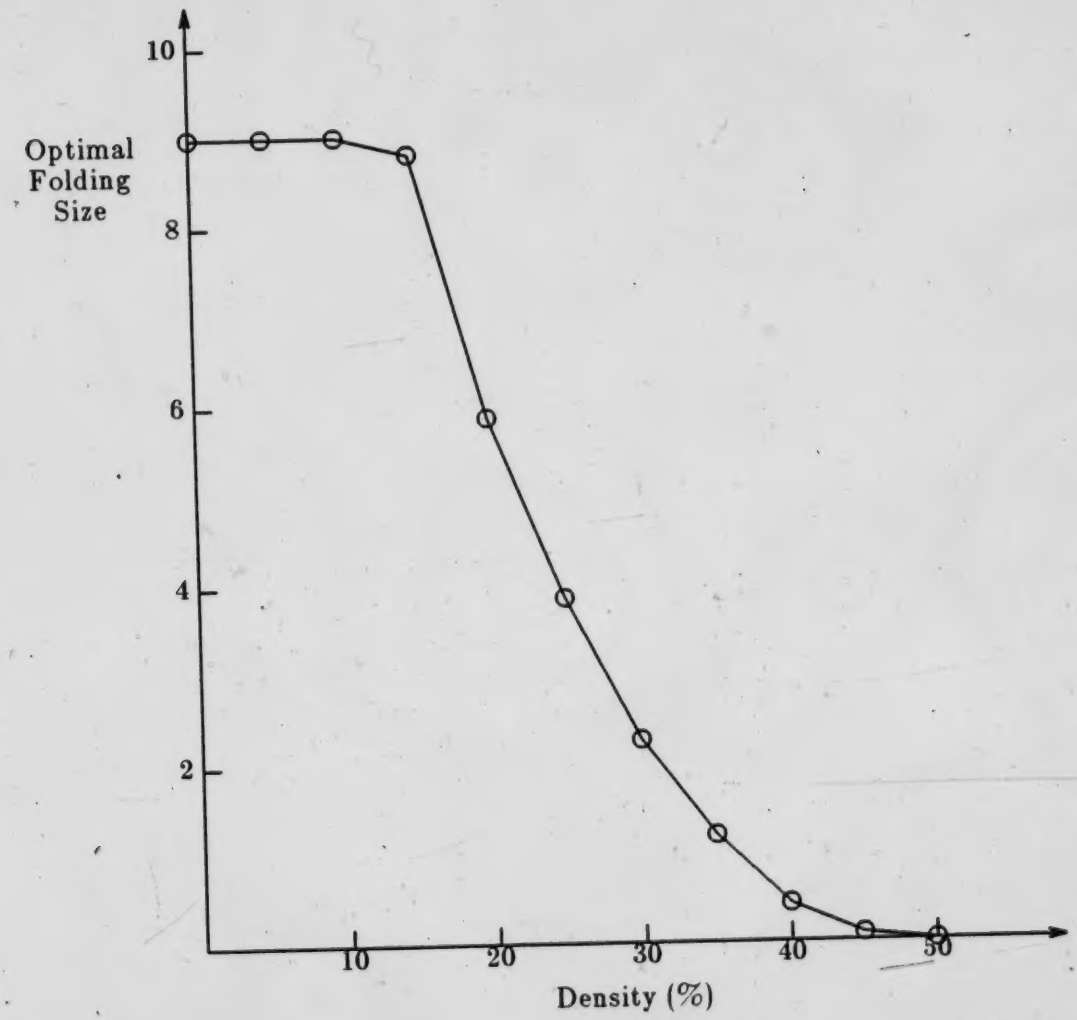

Figure 5.1 Optimal Folding Size vs Density

for PLAs of size r = 20 and c = 18

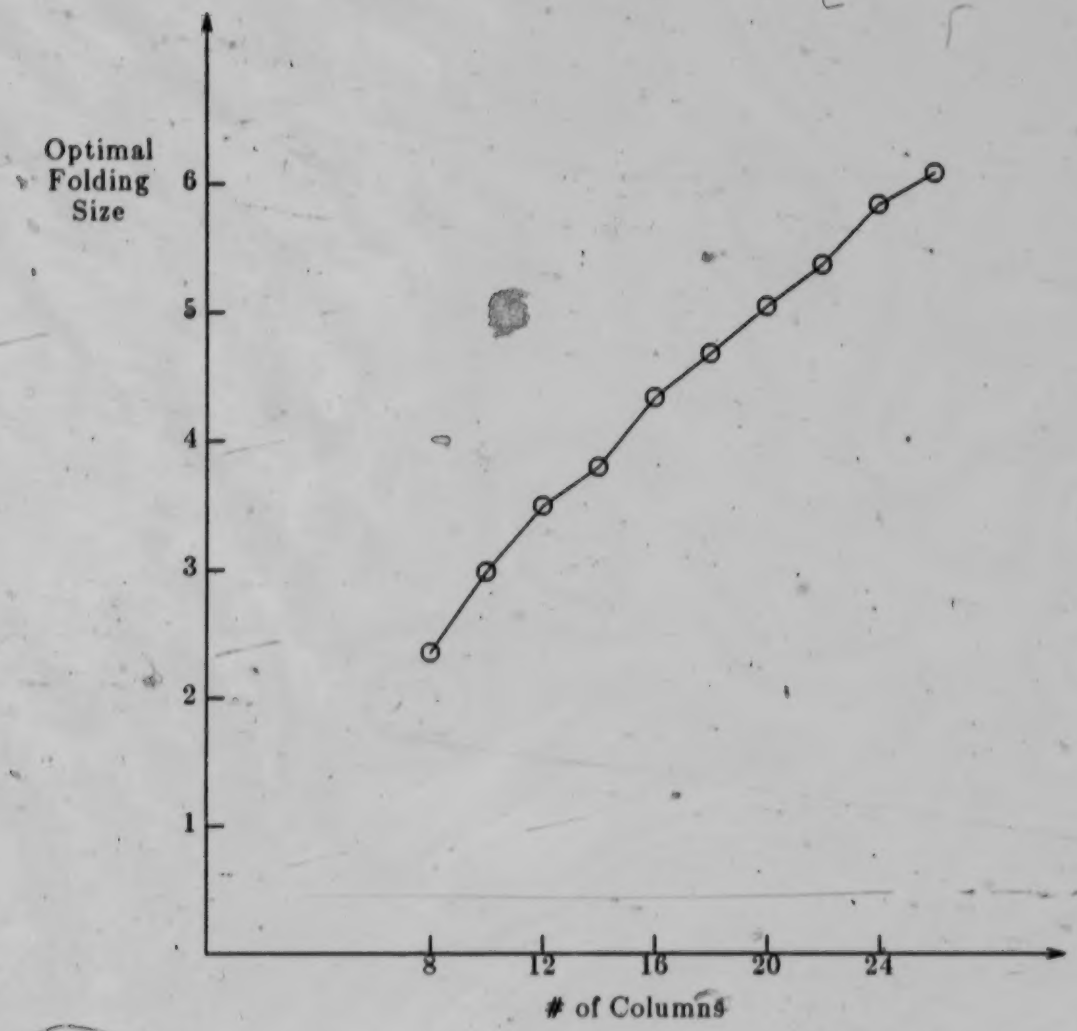

Figure 5.2 Optimal Folding Size vs. # of Columns
for PLAs of size r = 16 and density = 25%

5.4. Bounding Measures

The basic difference between the various branch and bound algorithms used for optimal PLA folding is the method used to determine the upper limit of the size of the maximum folding set achievable from a specified position in the search space. Comparison of these different algorithms is difficult. Results are given for specific PLAs and it is impractical to publish the location of all active intersections for all of the PLAs with which a folding algorithm was tested.

To overcome this problem, two measures are introduced. First, the *Class Bounding Measure* is defined as follows.

$$M_{class} = \frac{\left(T_{ALG}\right)^{2/C}}{\left(T(C)\right)^{2/C}}$$

This measure is based on the use of the *Basic Optimal Folding Algorithm* described earlier. The measure is derived from a large random sample of PLAs from a given class. A class of PLAs are all those of the same size and density and is represented by the 3-tuple <# of rows, # of columns, density>. Recall that the density of a PLA is the percentage of intersections within the PLA that are active. For example, the class <20,16,25%> represents all PLAs of size r = 20 and c = 16 having a density of 25%. T_{ALG} is the number of folding sets examined using the *Basic Optimal Folding Algorithm* without any bounding. T(C) is the total number of distinct folding sets possible as described earlier. The number of folding sets raised to the power $2/C$ gives an indication of the average width of the search tree if it were balanced. Recall, that in a balanced, uniform, full tree, the number of leaf nodes is

$$(\textit{width of the tree at each level})^{\textit{depth of the tree}}$$

The total number of distinct folding sets in the search tree is $T(C)$ as was derived earlier. The depth of the search tree is $C/2$ because there are at most $C/2$ column pairs in a folding set. Thus

$$T(C) = (\textit{average width of the tree at each level})^{C/2}.$$

Solving we get

$$\textit{average width of the tree at each level} = \left(T(C)\right)^{2/C}.$$

Note that this is not exactly true as $T(C)$ is the total number of nodes in the search tree as opposed to just the leaf nodes but for the purpose of providing a relative measurement, it works fine. This is because $T(C)$ is dominated by the leaf node count.

The same equation applies to the *Basic Optimal Search Algorithm* except that the number of folding sets examined in the search tree is T_{ALG}. T_{ALG} is usually considerably less than $T(C)$, as many folding sets are eliminated using the result of Theorem 5.1. The ratio of the average tree widths varies depending on how many folding sets are eliminated by the *Basic Optimal Folding Algorithm* and this ratio represents the relative difficulty in finding the optimal folding set for a given class of PLAs. M_{class} varies from 0 to 1 with values near 0 indicating ease in obtaining the optimal folding and values near one indicating difficulty. In fact, any PLA class with a M_{class} greater than 0.3 usually requires considerable searching to find an optimal folding. A value of exactly one indicates that the *Basic Optimal Folding Algorithm* examined all $T(C)$ distinct folding sets. The value one exists only as a theoretical limit and would never occur in practice.

Table 5.2 shows M_{class} for PLAs of a fixed size and varying densities. The

values in the Table were empirically derived by optimally folding hundreds of PLAs. One observation to note, is that M_{class} is highest and thus optimal foldings are most difficult to find when the average optimal folding set size is slightly less than a complete folding where all columns are folded. This is due in part to the fact that the search for an optimal folding ends when a complete folding is found.

The second measure, called the *Bounding Comparison Measure*, is used for comparing different PLA optimal folding algorithms and is defined as follows.

$$M_{comp} = \frac{\#\ of folding\ sets\ examined\ without\ bounding}{\#\ of folding\ sets\ examined\ using\ bounding\ technique}$$

for a large random sample of PLAs in a given class. This measure also assumes the use of the *Basic Optimal Folding Algorithm* and thus compares results that are dependent upon qualities of the bounding technique, eliminating the effects of properties of specific PLAs. The number of folding sets examined while using the bounding technique includes any sets examined during the bounding process itself.

As an example, Table 5.2 shows M_{comp} for the branch and bound algorithm presented here for PLAs of a fixed size and varying densities. Each row inthe table was derived derived by optimally folding 200 random PLAs generated by the method described in the appendix. The results can be used by other authors to compare the performence of this branch and bound algoriths to others. It may be found that this algorithm performs well for some PLA classes and poorly for others. M_{comp} will help to show the strengths and weaknesses of each optimal folding algorithm. The most

effective technique may be a combination of all current methods.

Note that in the table, the computationally easiest PLAs to fold, as represented by M_{class}, are the ones that are either very dense, or very sparse. Dense PLAs are computationally easy to fold because very little folding occurs and thus the algorithm does not have to examine as many folding sets. Sparse PLAs are computationally easy to fold because the algorithm very quickly finds a perfect folding of $\frac{c}{2}$ folds and then quits. It appears that the toughest PLAs to optimally fold are the ones whose optimal folding size is slightly less than a perfect folding. In terms of actual computation time, PLAs of size r = c = 20 can take up to an hour of cpu time to optimally fold on a Vax 11/780.

PLA density	M_{class}	M_{comp}	avg opt. folding size
.10	.111	1.04	7.0
.15	.316	3.65	6.8
.20	.264	2.18	4.9
.25	.160	1.34	3.2
.30	.113	1.04	1.8

Table 5.2 Bounding Measures for PLAs of size r = 20, c = 14

Chapter 6

A Theoretical Analysis of PLA Folding

Current understanding of the PLA folding problem is limited to simple empirical evidence gained from studies using heuristic and branch and bound methods. Very little theoretical work has been done other than that found in [LVV82, RaL84]. No statistical analysis of PLAs or their properties is available. All of the results presented in the literature are based on small numbers of individual PLAs, thus no theoretical or statistical basis for comparisons of methods or results exist. This chapter presents a theoretical approach to the problem through an analytical and statistical analysis.

As a first step, what is needed is a method to estimate the foldability of a given PLA prior to attempting to actually fold it. This estimate must be based on simple fundamental properties of the PLA. These properties should be easily measurable and the estimate should be theoretically significant.

Using a random selection heuristic as a basis, this chapter presents such an estimate in the form of a probability density function (PDF) of the expected number of folds for random PLAs. This PDF is analytically derived and is shown to satisfy the above criteria. It is derived in terms of the fundamental properties of a PLA, r - the number of rows, c - the number of columns and d - the density. The derivation is then extended to account for a secondary effect, h(x) - the distribution of the PLA density among the columns. This PDF can be used to help guide heuristic folding algorithms, specifically in helping to determine when to terminate the search for more folds. As well, it gives insight into how the fundamental properties of a PLA

affect its foldability. In the derivation of the probability density function a number of results are produced which provide useful insight into the nature of the PLA folding problem. These results form a solid framework for future work. Empirical data obtained from folding thousands of randomly generated PLAs are included to illustrate that the derived probability density function reasonably models the PLA folding problem.

6.1. Basic Definitions

A PLA can be represented by a *personality matrix*. A personality matrix is a matrix of ones and zeros representing the intersections within the PLA. A one in a given position in the matrix indicates an active intersection at the corresponding position in the PLA. Similarly, a zero indicates an inactive intersection.

It will be assumed for this study that only simple column folding will be considered and that all columns may fold with each other. Suppose there is a given PLA as represented by a personality matrix. For the purpose of the analysis, let us introduce the following definitions.

Let

r = the number of rows in the personality matrix.

c = the number of columns in the personality matrix.

and

d = *the density of the PLA*

$$= \frac{\text{\# of active intersections}}{r \times c}$$

Let d_i be defined as the ***column density*** of the i'th column. i.e.

$$d_i = \frac{\#\ of\ active\ intersections\ in\ column\ i}{r}$$

A PLA is said to have an "evenly distributed density" if

$$d_i = d_j$$

and

$$d = d_i$$

for all i,j where

$1 \leq i,j \leq c$

For PLAs that do not have an evenly distributed density, let h(x) be the distribution function of the column densities.

i.e

$h(x)$ = the probability that a given column will have exactly x active intersections.

It will be assumed for now that all PLAs have an evenly distributed density and the overall density d will represent d_i for all i. This restriction is removed in the last section of this chapter.

Figure 6.1 illustrates a PLA personality matrix and some of these definitions.

6.2. Deriving P_c

It is intuitively clear that if two columns are to be folded together, they must not have active intersections in any common rows. This follows from the set-theoretic definition of the problem where the inequality in Equation 3.1 is strictly less than. In terms of the model used by [HNS82b] and others, columns must be ***non-adjacent*** in order to fold. As an example, in Figure 6.1

	1	1	0	1	0	1	$r = 4$
	2	1	0	0	1	0	$c = 5$
row							
	3	0	1	1	0	0	$d = \frac{10}{20} = .5$
	4	0	1	0	1	1	
		1	2	3	4	5	$d_i = \frac{2}{4} = .5$ for all i
		col					

Figure 6.1 Evenly Dense PLA

---

columns 1 and 3 are not foldable because because they both have active intersections in row 1.

The function P_c is defined as follows:

If two columns are chosen at random, P_c is the probability that they are eligible for folding in that they do not share any common rows.

Theorem 6.1

P_c is given by the following formula:

$$P_c = \frac{\binom{r - d \times r}{d \times r}}{\binom{r}{d \times r}}$$

where d and r are the density and number of rows respectively as previously defined and the brackets indicate the combinatorial function which counts the number of combinations of n distinct objects.

Proof

Assume one column c_1 is chosen at random. A second column c_2 is chosen at random. c_2 contains $d \times r$ active intersections. Remember that the PLA has an evenly distributed density. These $d \times r$ active intersections can be placed in any of the r row positions. If the placement of the active intersections is viewed instead as a selection of $d \times r$ rows that are active, then it is clear that the total number of possible arrangements of c_2 is

$$\binom{r}{d \times r}$$

There are $r - d \times r$ rows which are not active in c_1. If a restriction is imposed that active intersections in c_2 must occur in rows that are inactive in c_1, then the total number of possible arrangements of c_2 now becomes

$$\binom{r - d \times r}{d \times r}$$

This restriction is simply that which allows column pairs to fold. Thus

$$P_c = \frac{\textit{total \# of arrangements eligible for folding}}{\textit{total \# of possible arrangements}}$$

$$= \frac{\binom{r - d \times r}{d \times r}}{\binom{r}{d \times r}} = \prod_{i=0}^{d \times r - 1} \frac{r - d \times r - i}{r - i} \qquad (6.1)$$

Q.E.D.

The following lemma confirms the observable result that if the PLA has more than 50% active intersections, then columns cannot fold.

Lemma 6.2

If $d > .5$ then no folding occurs.

Proof

By equation 6.1 in Theorem 6.1, if $d > .5$ then the numerator becomes zero. Thus the probability that two columns will fold is zero.

Q.E.D.

A plot of P_c vs d is shown in Figure 6.2 for $r = 20$. A Plot of P_c vs r is shown in Figure 6.3 for $d = .2$. These two graphs illustrate the relationship between the basic foldability of the columns of a PLA, the density, and number of rows. Note that P_c is independent of the number of columns in the PLA.

Theorem 6.3

For relatively large r and relatively small d, P_c can be approximated by

$$P_c approx = (1-d)^{d \times r}$$

Proof

Assume two columns c_1 and c_2 are chosen at random. There are $d \times r$ active intersections in c_1. For each of those active intersections the probability that the corresponding intersection in c_2 is inactive is $(1-d)$. The probability that all of the $d \times r$ corresponding intersections in c_2 are inactive is

$$(1-d)^{d \times r}$$

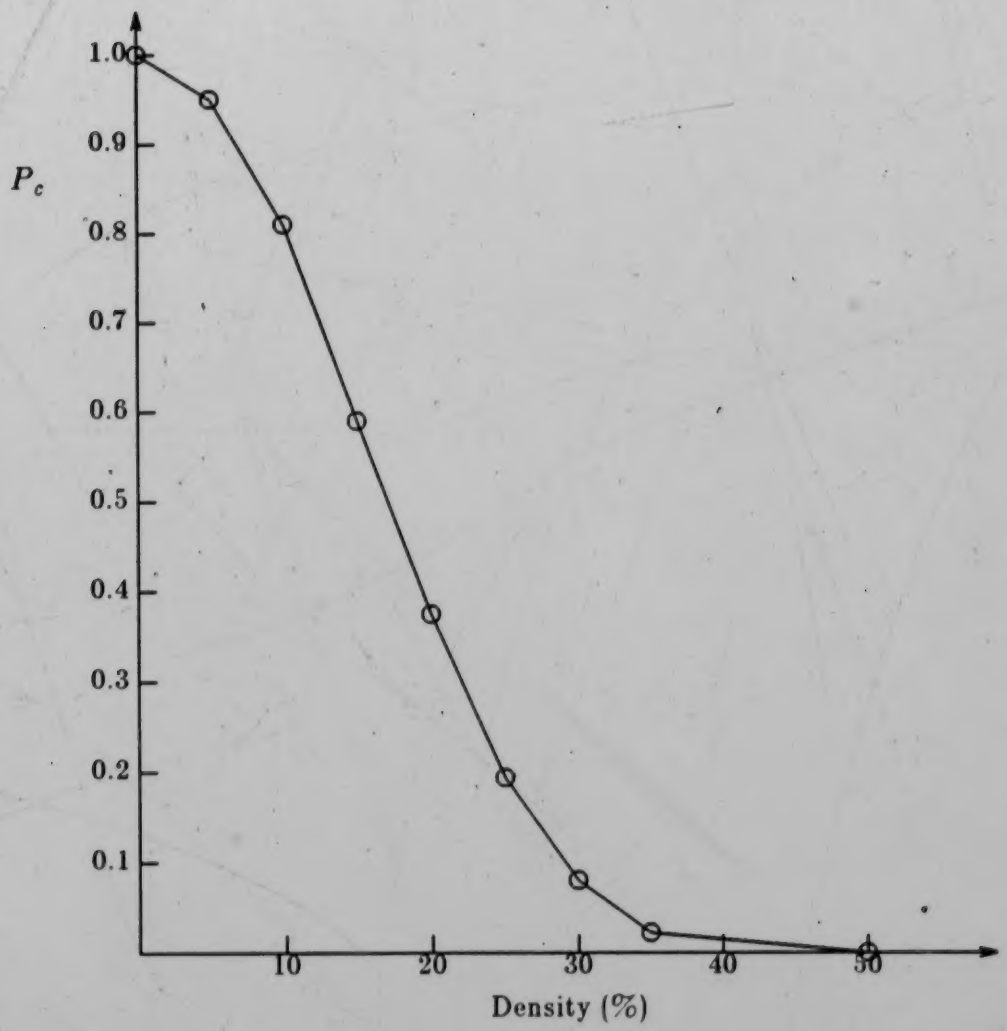

Figure 6.2 P_c vs density for r = 20

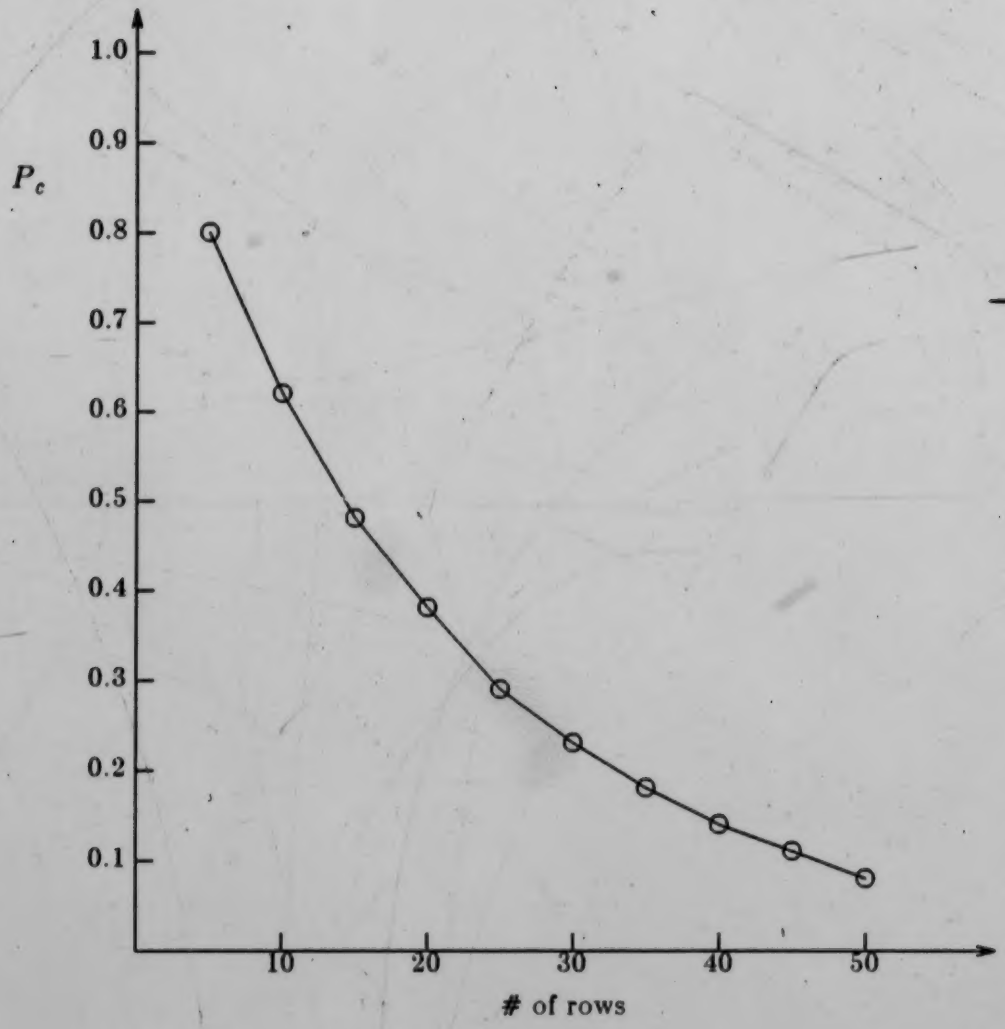

Figure 6.3 P_c vs r for density = 20%

If r is relatively large and d is relatively small, then it can be assumed that the probability that c_1 and c_2 intersect on a given row is independent of

whether or not they intersect on other rows and thus

$$P_c approx = (1-d)^{d \times r}$$

Q.E.D.

Figure 6.4 shows a comparison of $P_c approx$ and P_c vs d for $r = 20$. Note that in the Figure $P_c \leq P_c approx$.

Lemma 6.4

$$P_c \leq P_c approx$$

Proof

$$P_c = \frac{\binom{r-d\times r}{d\times r}}{\binom{r}{d\times r}} = \prod_{i=0}^{d\times r-1} \frac{r-d\times r-i}{r-i}$$

$$= \prod_{i=0}^{d\times r-1} \left[(1-d) - \frac{d\times i}{r-i}\right] \leq \prod_{i=0}^{d\times r-1} (1-d) = (1-d)^{d\times r}$$

Thus $P_c \leq P_c approx$

Q.E.D.

P_c is a fundamental measure of the foldability of a PLA. However it neglects certain constraints which will now be examined.

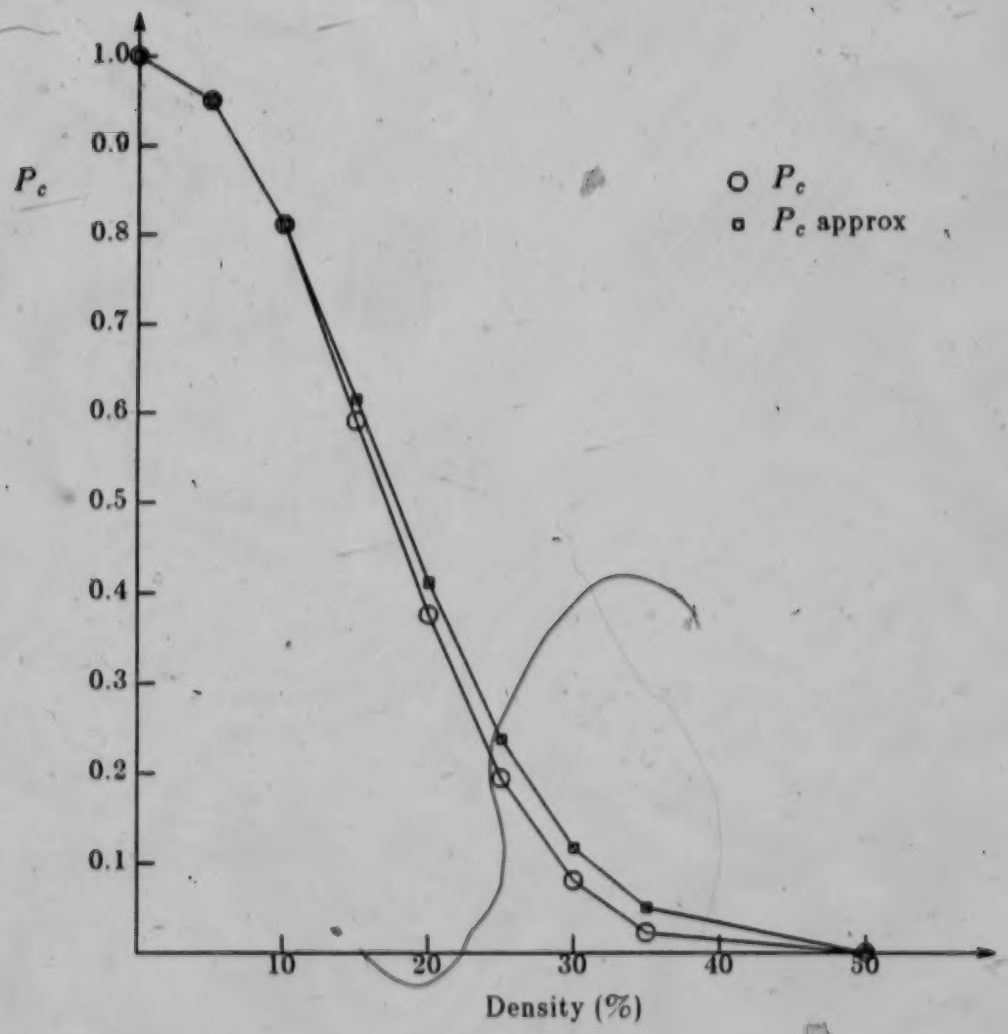

Figure 6.4 P_c and P_c *approx* vs density for r = 20

6.3. Expected Number of Folds

Constrained only by the condition that columns having active intersections on common rows are unfoldable, P_c is the probability that two randomly selected columns will fold. Based on this result, one would like to know the expected number of folds in the optimal folding of a PLA, of a given size and density, for a large sample of PLAs. This is a difficult problem and has not yet been solved analytically. Later in this chapter empirical data pertaining to this problem are presented.

An analytical solution to a less difficult version of this problem is presented here. First, the random selection heuristic is introduced for PLA folding.

Random Selection Heuristic

$F = \phi$; /* F is the folding set */
S = the set of all column pairs;
while ($S \neq \phi$) do
  randomly select a pair p_1 from S;
  $S = S - \{p_1\}$
  If (the column pair p_1 is foldable including considerations of constraints
    imposed by previous folds in F) then
    $F = F \bigcup p_1$
  endif;
endwhile;

When the algorithm terminates, F will contain a folding set for the PLA.

The *random selection folding size* of a PLA is defined to be the average

cardinality of F achieved using this algorithm for a large number of PLAs of the same size and density. Basically this algorithm is similar to the heuristic algorithm used in [HNS82b] but the selection process is random.

The basic problem can now be stated.

Problem

Determine the probability density function of the expected number of folds achieved using the random selection heuristic over a large sample of PLAs of fixed size and density.

In order to calculate the probability of achieving folding sets of a given size, one must first examine the factors that restrict folding. There are three distinct causes of folding restrictions.

Restriction #1.

Columns can only fold with other columns with which they do not share common rows.

Restriction #2.

Columns can fold only once. If the folding set F contains the pair (c_a, c_b), then column c_a and column c_b are now considered ineligible for folding with any other column.

Restriction #3.

Column pairs can fold only if they do not violate any of the partial orders that have been imposed on the rows by previous folds. For example, consider the PLA in Figure 8.1. Note that

column c_1 and c_2 can fold and

column c_3 and c_4 can fold

but both cannot fold together because of the ordering of the rows.

These three restrictions will be the basis of the derivation of the probability density function. Restriction #1 has already been addressed by the previously derived P_c.

For restriction #2, the function S(n) is introduced and defined as follows.

S(n) is the number of folding pairs available for selection after n previous folds have occurred based only on restriction #2. That is, it is the number of folding pairs which contain none of the $2\times n$ columns used in the n folding pairs in F.

Theorem 6.5

$$S(n)=\left(c-2\times n\right)\times\left(c-2\times n-1\right)$$

where n = the number of previous folds and c = the number of columns.

Proof

Initially, since no previous folds have been made, all column pairs are available. Thus,

$$S(0)=c\times(c-1)$$

After the first fold is made two columns are now no longer available, leaving c-2 columns. This would result in

$$S(1)=(c-2)\times(c-3)$$

available column pairs.

In general, after n folds, there will be $c-2\times n$ columns left unfolded and thus

$$\left(c-2\times n\right)\times\left(c-2\times n-1\right)$$

available column pairs.

Therefore

$$S(n)=\left(c-2\times n\right)\times\left(c-2\times n-1\right)$$

Q.E.D.

S(n) represents the size of the pool of column pairs from which random selection can be made. Note that some of the column pairs in this pool may prove to be unfoldable because of restriction #1 or #3.

Let restriction #3 be ignored for the moment. It will be assumed that there is no interaction between different folding pairs in F. This of course, is an improper assumption, but it will be removed later in this chapter.

Lemma 6.6

The probability that for a given PLA the random selection heuristic will not be able to find any folds is given by

$$(1-P_c)^{c\times(c-1)}$$

$$=(1-P_c)^{S(0)}$$

Proof

Given that P_c is the probability that two randomly selected columns will fold, $1-P_c$ is the probability that two randomly selected columns will not fold. There are $c\times(c-1)$ possible folding pairs. Assuming that they are independent, the probability that none will fold is

$$(1-P_c)^{c\times(c-1)}$$

$$=(1-P_c)^{S(0)}$$

Q.E.D.

Let us now introduce a new function which will be referred to as B(n); B(n) is defined as the probability that the random selection heuristic will find *at least* n folds.

Lemma 6.7

B(1) , the probability that at least one fold will be found, is given by

$$B(1) = 1-(1-P_c)^{S(0)}$$

Proof

Remember that restriction #3 is relaxed.

B(1) = 1 - the probability that no folds will be found

but Lemma 6.6 states the probability that no folds will be found is

$$(1-P_c)^{S(0)}$$

so

$$B(1) = 1-(1-P_c)^{S(0)}$$

Q.E.D.

Theorem 6.8

$$B(n) = B(n-1)\times\left[1-(1-P_c)^{S(n-1)}\right]$$

Proof

B(n) = the probability that the random selection heuristic will find at least (n-1) folds *times* the probability that, given that it has found n-1 folds, it will find at least one more. The probability that it will find at least n-1 folds is B(n-1).

After n-1 folds have been found, there are S(n-1) pairs eligible for folding based on restriction #2 (see Theorem 6.5). The probability that any given one of those pairs will not fold because of restriction #1 is $(1-P_c)$.

The probability that none will be able to fold is $(1-P_c)^{S(n-1)}$. Thus, the probability that at least one of them will fold is 1 minus the probability that none will fold

$$= \left[1-(1-P_c)^{S(n-1)}\right]$$

so

$$B(n) = B(n-1)\times\left[1-(1-P_c)^{S(n-1)}\right]$$

Q.E.D.

Lemma 6.9

$$B(n) = \prod_{i=0}^{n-1}\left[1-(1-P_c)^{S(i)}\right]$$

Proof

This is easily derivable from the result of Theorem 6.8.

Q.E.D.

Now that B(n) is derived, let us now define the function C(n).

C(n) = the probability that the random selection heuristic will find *exactly* n folds.

Lemma 6.10

$$C(n) = B(n)-B(n+1)$$

Proof

The probability that *exactly* n folds occur is the probability that *at least* n

folds occur minus the probability that *more than* n folds occur.

Q.E.D.

Theorem 6.11

$$C(n) = (1-P_c)^{S(n)} \times \prod_{i=0}^{n-1} \left[1-(1-P_c)^{S(i)}\right]$$

Proof

$$C(n) = B(n) - B(n+1)$$

$$= \prod_{i=0}^{n-1} \left[1-(1-P_c)^{S(i)}\right] - \prod_{i=0}^{n} \left[1-(1-P_c)^{S(i)}\right]$$

$$= \left[\prod_{i=0}^{n-1} \left[1-(1-P_c)^{S(i)}\right]\right] \times \left[1-\left[1-(1-P_c)^{S(n)}\right]\right]$$

$$= (1-P_c)^{S(n)} \times \prod_{i=0}^{n-1} \left[1-(1-P_c)^{S(i)}\right]$$

Q.E.D.

Theorem 6.11 is the basic form of the probability density function for the random selection heuristic. However, recall that up to this point restriction #3 has been relaxed. This will now be included.

6.4. Row orders - Restriction #3

Recall that restriction #3 is the restriction that currently folded column pairs place on future folds during the heuristic folding process. This restriction is in the form of partial orders imposed upon the rows. This is illustrated by the folded PLA in Figure 6.5. The current folds in the figure impose the indicated orders upon the rows. Columns e and f would be foldable except for the fact that either orientation (column e folded above

column f or column f folded above e) would violate at least one of these row orders.

The partial orders of the rows are divided into two classes, ***primary*** and ***secondary***. Primary orders are those which exist directly because of previous individual folding pairs. In Figure 6.5 the first two row orders are primary ones.

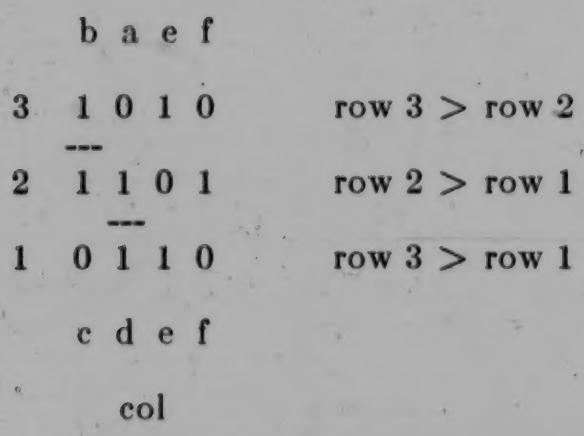

Figure 6.5 Row Orders Implied by Previous Folds

However, because of transitivity, the last order "row 3 > row 1" is also created. This is a secondary order as it results from a combination of the two primary orders.

In this study, only primary are considered. This introduces a slight inaccuracy into the calculations, however analysis becomes very difficult if composite orders are included. Our experience with empirical data indicates that secondary orders do not have a significant contribution to the final formula. The problem of representing the secondary orders analytically is non-trivial.

Suppose a PLA has r rows and a density d. The average number of active intersections per column is $d \times r$. Each fold will impose $(d \times r) \times (d \times r)$ partial orders onto the rows. Note that some of these partial orders may be identical to those produced by other folds. As an example, consider the PLA in Figure 6.1. Each fold produces $(.5 \times 4) \times (.5 \times 4) = 4$ partial row orders. Folding columns 1 and 2 produces the 4 partial orders

row 1 > row 3

row 1 > row 4

row 2 > row 3

row 2 > row 4

There are, for a given size of PLA, a total of $r \times (r-1)$ possible partial orders. The set R_F is defined as follows.

If F is a set of column folding pairs, then associated with the set F is a set R_F of primary partial orders imposed by the column folding pairs in F. R_F is a subset of the $r \times (r-1)$ possible partial orders.

$Q(n)$ is defined as follows.

$Q(n)$ is the average size of R_F, for F of size n.

In other words, $Q(n)$ is the average number of partial orders imposed on the rows after folding n column pairs. It is expressed as a percentage of the total $r \times (r-1)$ possible partial orders. Remember that only primary partial orders are considered. Note also that the limit on $Q(n)$ is 0.5. This is because it is not possible to have both partial row orders i>j and j>i in R_F

Lemma 6.12

$$Q(n) = 1 - \left(1 - \frac{d^2 \times r}{r-1}\right)^n$$

Proof

$Q(0)$ is, of course, zero. On average each fold produces $d^2 \times r^2$ partial orders so

$$Q(1) = \frac{d^2 \times r^2}{r \times (r-1)} = d^2 \times \frac{r}{(r-1)}$$

The second fold will also produce $d^2 \times r^2$ partial orders but a fraction of these will be repeats that already exist in R_F from the first fold. This fraction is proportional to the number of existing partial orders in R_F. However, $Q(1)$ represents this fraction of partial orders that already exist.

This means that $\left\{1 - Q(1)\right\}$ represents the fraction of partial orders that do not already exist and will be new.

Thus on average, $\left(1 - Q(1)\right) \times d^2 \times r^2$ new orders will be created.

This represents $\left(1 - Q(1)\right) \times \frac{d^2 \times r^2}{r \times (r-1)}$ new orders when expressed in terms of the fraction of the total $r \times (r-1)$ partial orders.

Thus,

$$Q(2) = Q(1) + \left(1 - Q(1)\right) \times d^2 \times \frac{r}{(r-1)}$$

In general,

$$Q(n) = Q(n-1) + \left(1 - Q(n-1)\right) \times d^2 \times \frac{r}{(r-1)}$$

Solving this recurrence produces

$$Q(n) = 1 - \left(1 - \frac{d^2 \times r}{r-1}\right)^n$$

Q.E.D.

So Q(n) represents the average number of existing partial row orders after n folds have taken place. It is in fact only approximate as it neglects secondary row orders. Remember, it is expressed in terms of a percentage of the total $r\times(r-1)$ possible partial orders.

The function V(n) is defined as follows.

V(n) is the probability that a randomly chosen column folding pair will be ***unconstrained*** by partial orderings in R_F, given there are n previous folds in F.

Theorem 6.13

$$V(n) = \prod_{i=0}^{d^2\times r^2-1} \frac{r\times(r-1)-Q(n)\times r\times(r-1)-i}{r\times(r-1)-i}$$

Proof

The total number of possible partial row orders is $r\times(r-1)$. Assume that n folds have been made. The n+1'st fold produces $d^2\times r^2$ partial orders. The total number of possible sets of $d^2\times r^2$ partial orders is

$$\binom{r\times(r-1)}{d^2\times r^2}$$

There are already existing $Q(n)\times r\times(r-1)$ partial orders in R_F from the n previous folds. For each partial order in R_F, there is a corresponding partial order which is disallowed in future folds. For example, if

Row 6 > Row 2

is a partial order in R_F, then the partial order

Row 2 > Row 6

is disallowed in future folds. Any future folds that produce this partial order

would be *constrained* by partial orders and would not be allowed to fold. Thus, after n folds have occured, there are $Q(n)\times r\times(r-1)$ partial orders that are not allowed to occur. This means there are

$$r\times(r-1)-Q(n)\times r\times(r-1)$$

partial orders which are allowed to occur. If the $d^2\times r^2$ new partial orders resulting from the n+1'st fold are to not cause any conflict, then they must be restricted to this set of allowable partial orders. The number of possible arrangements of these new partial orders which adhere to this restriction is

$$\begin{pmatrix} r\times(r-1)-Q(n)\times r\times(r-1) \\ d^2\times r^2 \end{pmatrix}$$

The probability that a randomly selected column pair will be unconstrained by previous partial orders is (the number of ways of arranging the $d^2\times r^2$ new partial orders so that they do not cause conflict) divided by (the total number of ways of arranging the $d^2\times r^2$ partial orders).

Thus

$$V(n)=\frac{\begin{pmatrix} r\times(r-1)-Q(n)\times r\times(r-1) \\ d^2\times r^2 \end{pmatrix}}{\begin{pmatrix} r\times(r-1) \\ d^2\times r^2 \end{pmatrix}}$$

$$=\prod_{i=0}^{d^2\times r^2-1}\frac{r\times(r-1)-Q(n)\times r\times(r-1)-i}{r\times(r-1)-i}$$

Q.E.D.

This now allows the updating of Theorem 6.8. Remember that B(n) is the probability that the random selection heuristic will find n or more folds.

Lemma 6.14

With the partial orders that are imposed on the rows (i.e. Restriction #3) now being considered:

$$B(n) = B(n-1)\times\left[1-\left(1-P_c\times V(n-1)\right)^{S(n-1)}\right]$$

Proof

The argument matches that of Theorem 6.8 with some additions.

B(n) = the probability that the random selection heuristic will find at least (n-1) folds *times* the probability that, given that it has found n-1 folds, it will find at least one more. The probability that it will find at least n-1 folds is B(n-1). After n-1 folds have been found, there are S(n-1) pairs eligible for folding based on restriction #2.

a) the probability that a given column pair is unconstrained because of active intersections on common rows is P_c

b) The probability that it is unconstrained because of partial row orders imposed by the n-1 previous folds is V(n-1).

The probability that it is unconstrained by both a) and b) is $P_c\times V(n-1)$.

The probability that it is constrained by either a) or b) is $1-P_c\times V(n-1)$.

The probability that all S(n-1) available column pairs will be constrained is

$$\left(1-P_c\times V(n-1)\right)^{S(n-1)}$$

The probability that at least one of them will fold is 1 minus the probability that none will fold, which is

$$1-\left(1-P_c\times V(n-1)\right)^{S(n-1)}$$

and so

$$B(n) = B(n-1)\times\left[1-\left(1-P_c\times V(n-1)\right)^{S(n-1)}\right]$$

Q.E.D.

Theorem 6.15

Including consideration of Restriction #3

$$B(n) = \prod_{i=0}^{n-1}\left[1-\left(1-P_c\times V(i)\right)^{S(i)}\right]$$

Proof

This is easily derivable from the results of Lemma 6.14.

Q.E.D.

Theorem 6.16

Including considerationof Restriction #3,

$$C(n) = \left(1-P_c\times V(n)\right)^{S(n)}\times\prod_{i=0}^{n-1}\left[1-\left(1-P_c\times V(i)\right)^{S(i)}\right]$$

Proof

Lemma 6.10 states

$$C(n) = B(n)-B(n+1)$$

Lemma 6.14 gives us an updated derivation of B(n) which can be substituted into the above equation. The equation is then reduced in a manner identical to that done in Theorem 6.11, with the new result

$$C(n) = \left(1-P_c\times V(n)\right)^{S(n)}\times\prod_{i=0}^{n-1}\left[1-\left(1-P_c\times V(i)\right)^{S(i)}\right]$$

Q.E.D.

This now allows us to state the major result.

Theorem 6.17

Let PDF(n,r,c,d) be the probability density function of the expected number of folds using the random selection heuristic. PDF(n,r,c,d) is the probability that exactly n folds will be found, given

r = number of rows

c = number of columns

d = density.

$$PDF(n,r,c,d) = \left(1 - P_c \times V(n)\right)^{S(n)} \times \prod_{i=0}^{n-1} \left[1 - \left(1 - P_c \times V(i)\right)^{S(i)}\right]$$

where

$$P_c = \frac{\binom{r - d \times r}{d \times r}}{\binom{r}{d \times r}}$$

$$V(n) = \prod_{i=0}^{d^2 \times r^2 - 1} \frac{r \times (r-1) - Q(n) \times r \times (r-1) - i}{r \times (r-1) - i}$$

$$S(n) = \left(c - 2 \times n\right) \times \left(c - 2 \times n - 1\right)$$

$$Q(n) = 1 - \left(1 - \frac{d^2 \times r}{r-1}\right)^n$$

Proof

The proof follows from the definition of C(n) and its formula in Theorem 6.16. PDF(n,r,c,d) is C(n).

Q.E.D.

A plot of PDF for random PLAs of 30 rows, 12 columns and density of 20% is shown in Figure 6.6. The plot illustrates that the function behaves as one would expect a probability density function to behave.

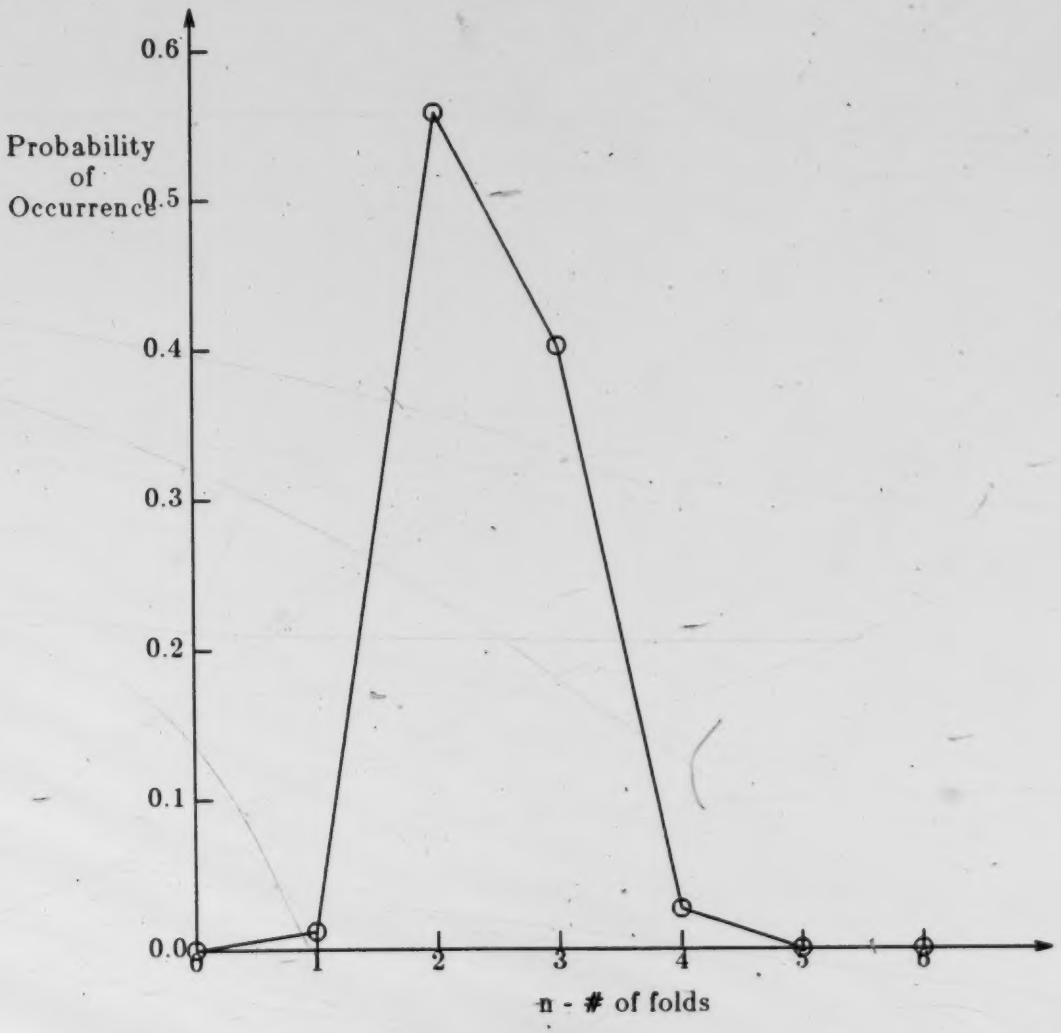

Figure 6.6 PDF(n,r,c,d) vs n as derived for PLAs of size r=30, c=12 and d=20%

Finally, to summarize the assumptions upon which the results so far are based.

1) folding is done using the random selection heuristic.

2) only simple column folding is allowed.

3) all PLAs are random and have evenly distributed densities.

4) only primary row orders are considered.

The empirical data in the next section will illustrate that, given these assumptions, the formulae reasonably model the PLA folding problem. The assumption of evenly distributed densities can be overcome by some modifications to the derivations as presented in the last section of this chapter.

6.5. Experimental Data

In this section experimental data from randomly generated PLAs are used to illustrate some of the analytical results that have been achieved. For the purpose of comparison, the randomly generated PLAs were restricted to those with properties matching the assumptions upon which the derivations of formulae in the last section were based. Specifically, only PLAs with evenly distributed column densities were used. The program used to generate the random PLAs is provided in the Appendix of this thesis. Actual PLAs used in industrial applications may have properties that affect foldability and are not modelled by the formulae.

The function P_c is examined first. Recall that P_c is the probability that two columns are foldable in that they do not have active intersections in any common rows. The derived formula for P_c is given in Theorem 6.1. One

thousand randomly generated PLAs were examined to measure P_c and to compare the results with the formula. Figure 6.7 shows a plot of P_c "measured" vs P_c "derived" for PLAs of size 20 by 16 and for varying densities.

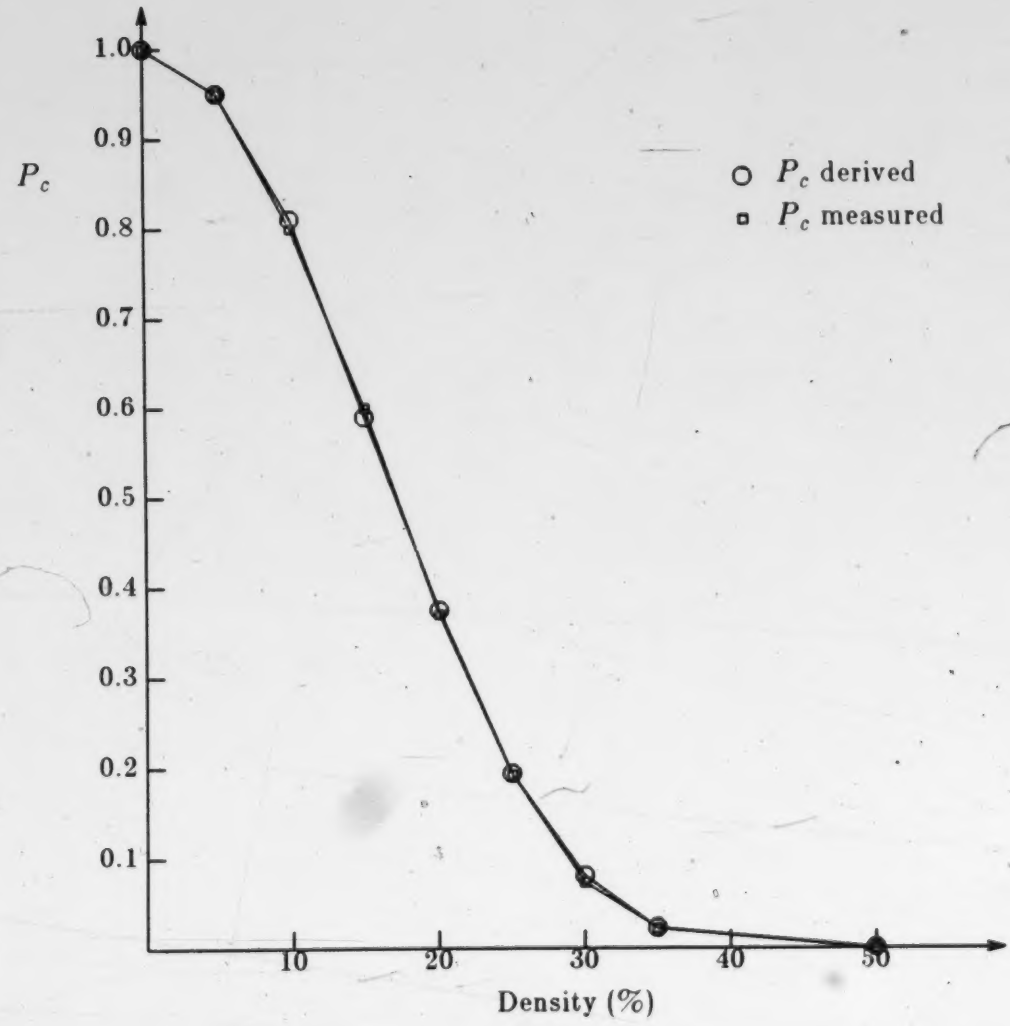

Figure 6.7 P_c derived and P_c measured vs d
for PLAs of size r = 20

As one would expect, the two functions almost exactly overlap.

The function S(n) is an exact combinatorial function and requires no verification.

Q(n) is defined as the average size of R_F for a folding set F of size n. Recall that R_F is the set of row orders imposed on future folds by the n previous folds as the random selection heuristic progresses. Lemma 6.12 gives the formula for Q(n) and it is represented as a percentage of all possible row orders. The formula only considers restrictions imposed by *primary* row orders. Figure 6.8 plots the derived values of Q(n) for r = 20, c = 16 and d = 20% over a range of n. Superimposed in the same figure is a plot of Q(n) values as measured from a large sample of PLAs with the same parameters. Note that the measured Q(n) is is always larger than the derived value because it considers secondary row orders as well as primary ones. This difference does not have a significant effect on the final result.

The final probability density function is given in Theorem 6.17. Figure 6.9 shows a plot of this function for r = 30 c = 12 and density = 20%. Superimposed on this graph is the probability density function as measured from 600 randomly generated PLAs of the same size and density, and with a constant column density distribution.

The graphs presented in this section illustrate that the analytical formulae do reasonably match the behavior of the randomly generated PLAs, under the given assumptions. Similar matchings occur for PLA parameters other than the specific examples selected for these graphs.

There may be certain characteristics exhibited by industrial PLAs that

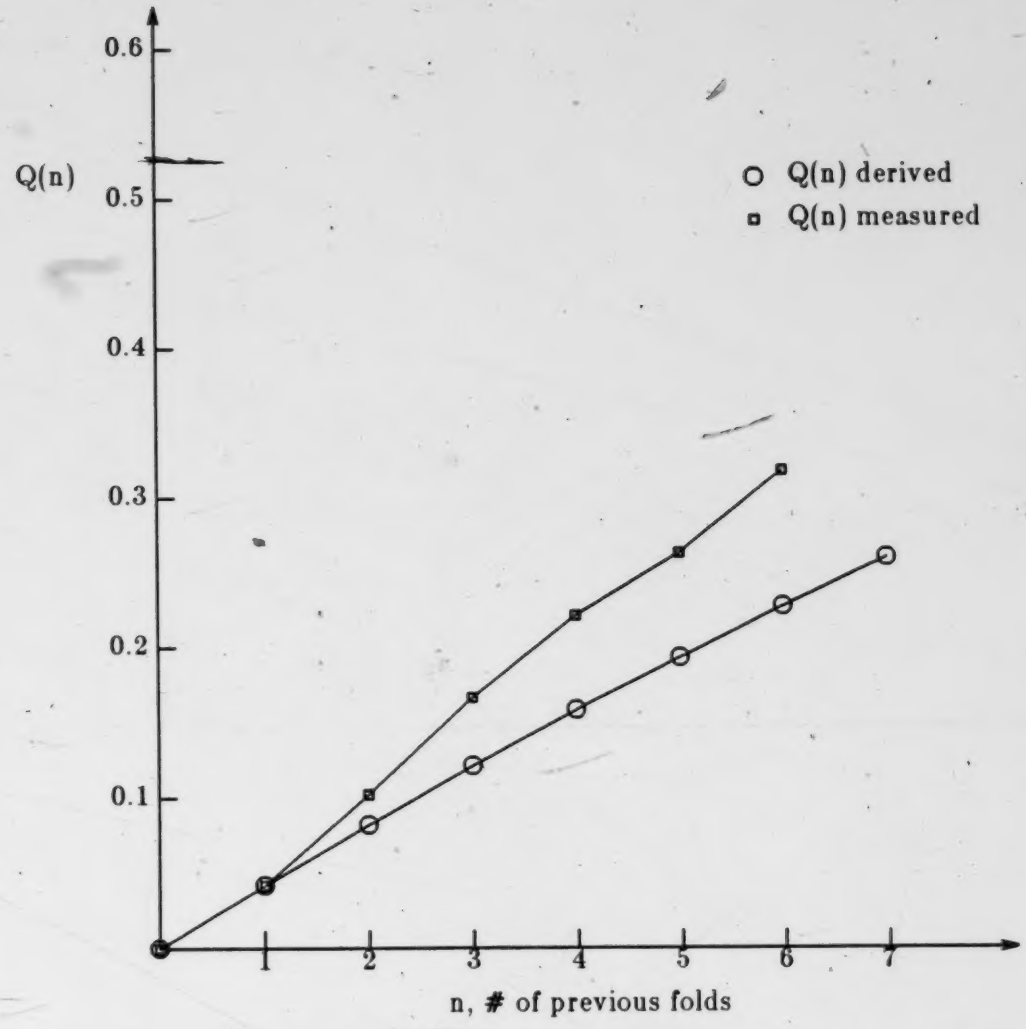

Figure 6.8 Q(n) derived and Q(n) measured
for PLAs of size r=20, c=16 and d=20%

differ from those exhibited in the evenly distributed, random PLAs used in this study. As an example, practical PLAs often have one or two columns which contain almost all active intersections, corresponding to a reset line. This can cause a substantial increase in the density of the PLA but does not

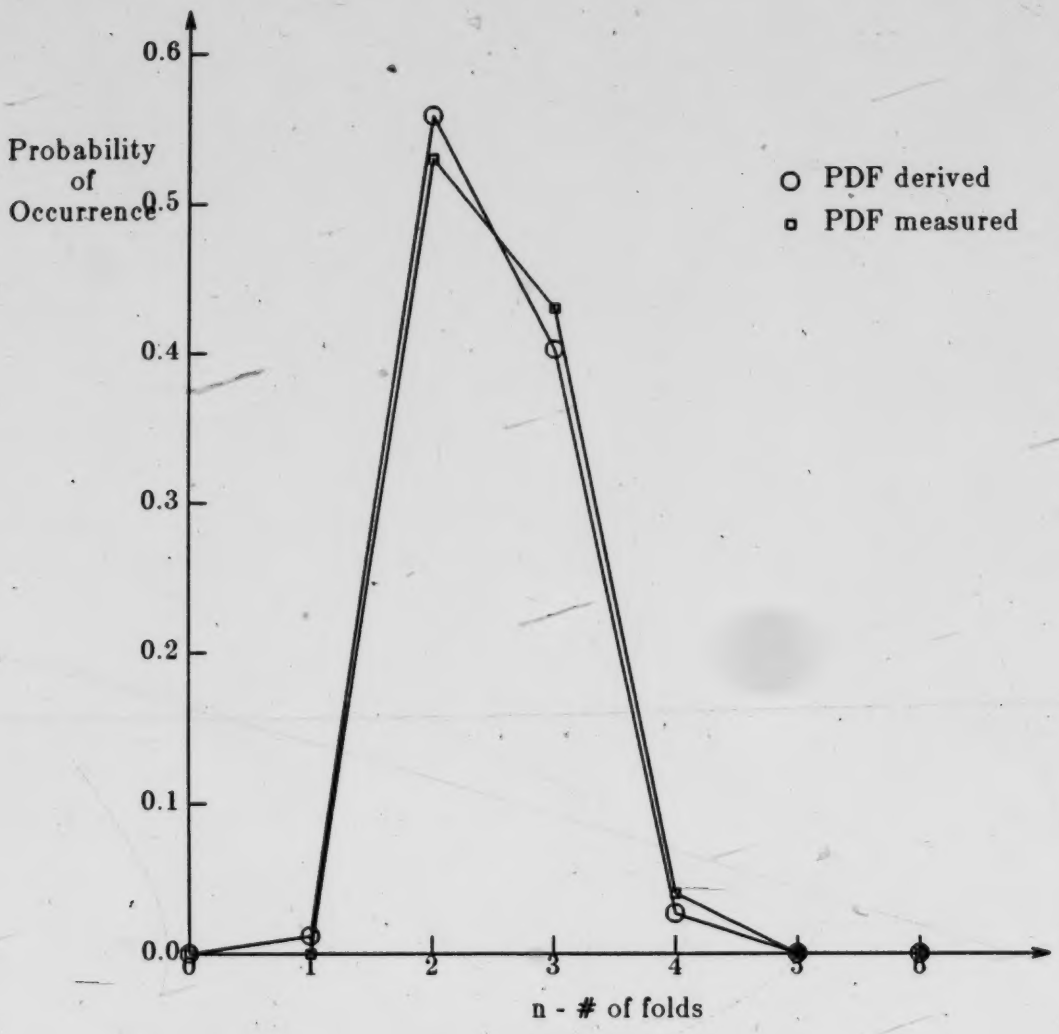

Figure 6.9 PDF derived and PDF measured
for PLAs of size r = 30, c = 12 and d = 20%

result in a corresponding substantial reduction in foldability as only one or two columns are affected. In addition, logic minimization techniques may affect the foldability of the resulting PLA in some subtle way. For instance, most logic minimization programs will eliminate identical rows and columns.

The effects of such actions are not obvious and depend upon the minimization techniques used. Industrial PLAs can have up to 100 separate rows or columns. Further research is needed to determine the unique characteristics of industrial PLAs and their effect on foldability.

Finally, one would expect the empirical results to differ, if instead of using the Random Selection Heuristic, optimal folding was performed. Figure 6.10 illustrates such a comparison for PLAs of size r = c = 16 and density = 25%. The empirically measured PDF is shown for both the Random Selection Heuristic and optimal folding. It can be seen that the shape of the two PDFs are similar, but the optimal folding achieves more folds. This is as one would expect. In the next chapter, the Random Selection Heuristic will be extended to achieve a PDF arbitrarly close to the PDF of optimal folding.

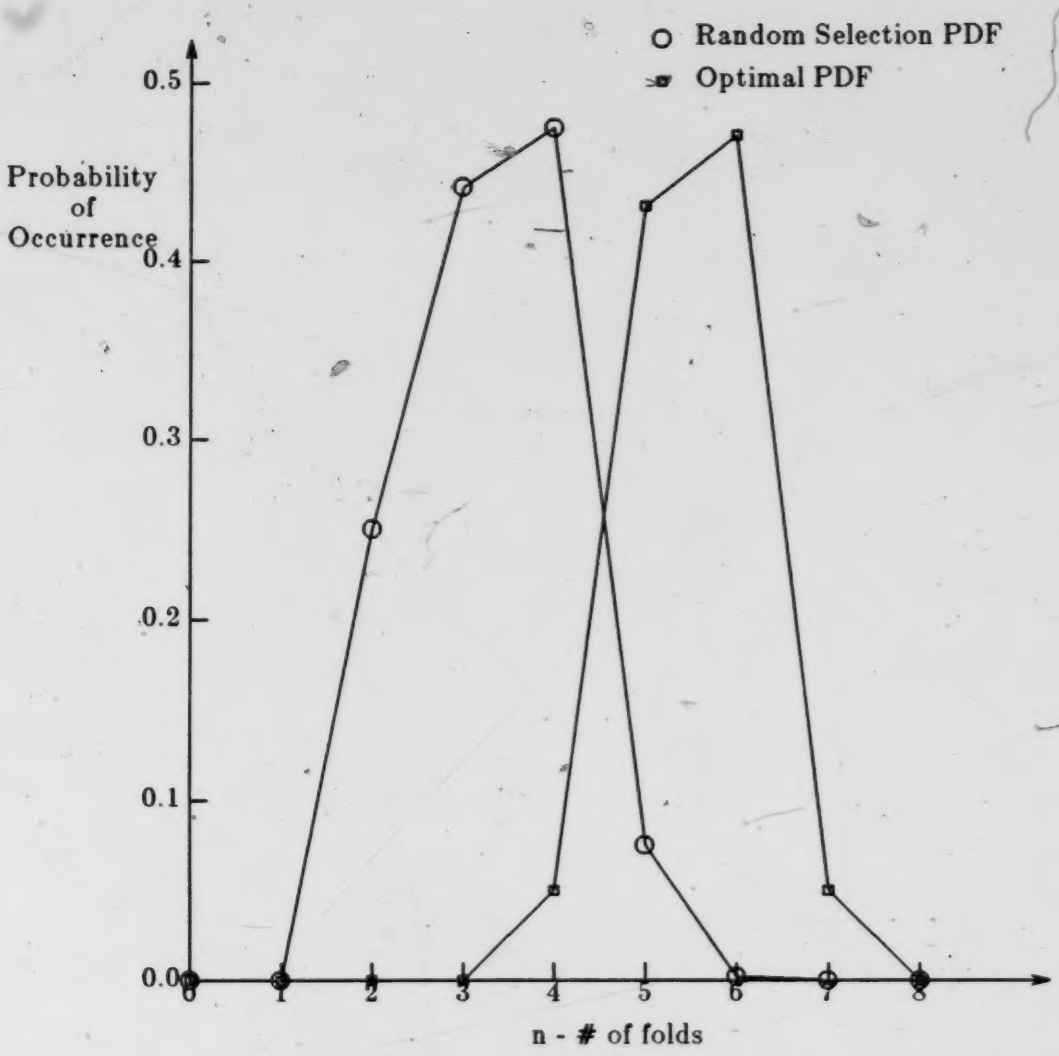

Figure 6.10 PDF for Random Selection Heuristic and Optimal Folding for PLAs of size r = 16, c = 16 and d = 25%

6.6. Removing the Evenly Distributed Density Assumption

Theorem 6.17 was derived under the assumption that all of the columns in the PLA have the same density. It is possible to generalize the result to cover PLAs that do not have this property. Recall that, for a set of PLAs, h(x) is the distribution function of the column densities.

i.e

$h(x)$ = the probability that a given column will have exactly x active intersections.

This function h(x) affects the derivation of the final probability density function in a number of ways. Basically the column density can no longer be represented as a constant d, for all columns, but rather must be represented as a random variable which has a distribution represented by h(x). First we revise the derivation of P_c.

6.6.1. Revising P_c

Theorem 6.18

With evenly distributed column density no longer being assumed,

$$P_c = \sum_{a=1}^{r} \sum_{b=1}^{r} \left[h(a) \times h(b) \times \left(\prod_{i=0}^{a-1} \frac{r-b-i}{r-i} \right) \right]$$

Proof

The proof follows that of Theorem 6.1 with the modification that the columns c_1 and c_2 no longer have the same density. They can have a variety of densities with a distribution represented by h(x).

Let a and b represent the number of active intersections in columns c_1 and

c_2 respectively. Let d_1 and d_2 represent the density of columns c_1 and c_2 respectively. Thus

$$d_1 = \frac{a}{r} \text{ and } d_2 = \frac{b}{r}$$

Now following the arguments in Theorem 6.1, we can derive a new function, $P_{spec}(d_1, d_2)$, as the probability that two columns with density d_1 and d_2 will not share common rows.

$$P_{spec}(d_1, d_2) = \frac{\binom{r - d_1 \times r}{d_2 \times r}}{\binom{r}{d_2 \times r}} = \prod_{i=0}^{d_1 \times r - 1} \frac{r - d_2 \times r - i}{r - i}$$

Substituting for d_1 and d_2 we get

$$P_{spec}(d_1, d_2) = \prod_{i=0}^{a-1} \frac{r - b - i}{r - i}$$

The probability that column c_1 will have a active intersections and column c_2 will have b active intersections is $h(a) \times h(b)$.

P_c can be derived by averaging the the value of $P_{spec}(d_1, d_2)$ over the entire distribution of possible values of a and b. Thus

$$P_c = \sum_{a=1}^{r} \sum_{b=1}^{r} \left[h(a) \times h(b) \times \left(\prod_{i=0}^{a-1} \frac{r - b - i}{r - i} \right) \right]$$

Q.E.D.

6.6.2. Revising Row Orders

P_c is not the only function that is affected by a distribution of column densities. In Theorem 6.4 it was shown that, on average, $(d \times r) \times (d \times r)$ partial row orders are created by every folding pair. Let us define $PO_{h(x)}$ as the average number of partial row orders created by a folding pair (c_1, c_2) for columns with a density distribution represented by h(x).

Theorem 6.19

$$PO_{h(x)} = \sum_{a=1}^{r} \sum_{b=1}^{r} \left[h(a) \times h(b) \times \left(\frac{\prod_{i=0}^{a-1} \frac{r-b-i}{r-i}}{P_c} \right) \times a \times b \right]$$

Proof

Recall that a and b are the number of active intersections and d_1 and d_2 are the densities of the two columns c_1 and c_2. For each possible combination of a and b, the probability that that particular combination will occur, if randomly selected, is $h(a) \times h(b)$. However, there is another factor which affects the probability of c_1 and c_2 having a and b active intersections respectively. Columns with low densities (and thus a low number of active intersections) are more likely to fold and are thus more likely to be in the folding set. In order to account for this, we must calculate $PF(d_1, d_2)$, the probability that a random column pair, with given column densities d_1 and d_2, will be placed in the folding set by the Random Selection Heuristic. Recall that the Random Selection Heuristic selects a column pair to be included in the folding set by continually picking random column pairs until one is found which is foldable.

If a random column pair has column densities d_1 and d_2, the probability that the pair will fold and be placed in the folding set on the first selection is $P_{spec}(d_1,d_2)$.

The probability that the first selection will not fold and that it will be on the second selection that the pair will fold and be placed in the folding set is

$$\begin{aligned}&(\textit{the probability that the first selection will not fold})\times\\&(\textit{the probability that the random column pair will fold})\\&=(1-P_c)\times P_{spec}(d_1,d_2)\end{aligned}$$

Extending this: the probability that it will be on the i'th selection that the random column pair will fold and be placed in the folding set is

$$(1-P_c)^{i-1}\times P_{spec}(d_1,d_2)$$

Thus, $PF(d_1,d_2)$, the probability that a random column pair, with given column densities d_1 and d_2, will be placed in the folding set is

(the probability that it will be placed on the first selection)

+

(the probability that it will be placed on the second selection)

+

etc.

Thus,

$$\begin{aligned}PF(d_1,d_2)&=P_{spec}(d_1,d_2)\times\left[1+(1-P_c)+(1-P_c)^2+\cdots\right]\\&=P_{spec}(d_1,d_2)\times\left[\frac{1}{P_c}\right]\\&=\frac{P_{spec}(d_1,d_2)}{P_c}\\&=\frac{\prod_{i=0}^{a-1}\frac{r-b-i}{r-i}}{P_c}\end{aligned}$$

So, for a random column pair (c_1, c_2) with a and b active intersections respectively, (i.e. column densities of d_1 and d_2), the probability that the particular combination of a and b will occur *and* that the column pair will be placed in the folding set is

$$h(a) \times h(b) \times PF(d_1, d_2)$$

$$= h(a) \times h(b) \times \left[\frac{\prod_{i=0}^{a-1} \frac{r-b-i}{r-i}}{P_c} \right]$$

The number of partial row orders that a sepcific folding pair will create is $a \times b$ The formula $PO_{h(z)}$ simply averages this out over all combinations of a and b. Thus,

$$PO_{h(z)} = \sum_{a=1}^{r} \sum_{b=1}^{r} \left[h(a) \times h(b) \times \left(\frac{\prod_{i=0}^{a-1} \frac{r-b-i}{r-i}}{P_c} \right) \times a \times b \right]$$

Q.E.D.

We can now replace the term $(d \times r) \times (d \times r)$ with the term $PO_{h(z)}$ in the derivations of the functions Q(n) and V(n) in Theorems 6.12 and 6.13 respectively. This allows us to make the following extension to Theorem 6.17.

Theorem 6.20

pdf(n,r,c,d,h(x)) is the probability density function for expected number of folds using the random selection heuristic on random PLAs.

$$PDF(n,r,c,d,h(x)) = \left(1-P_c\times V(n)\right)^{S(n)}\times\prod_{i=0}^{n-1}\left[1-\left(1-P_c\times V(i)\right)^{S(i)}\right]$$

where

r = number of rows

c = number of columns

d = density.

h(x) = the distribution function of the column densities.

$$P_c = \sum_{a=1}^{r}\sum_{b=1}^{r}\left[h(a)\times h(b)\times\left(\prod_{i=0}^{a-1}\frac{r-b-i}{r-i}\right)\right]$$

$$V(n) = \prod_{i=0}^{PO_{h(x)}-1}\frac{r\times(r-1)-Q(n)\times r\times(r-1)-i}{r\times(r-1)-i}$$

$$S(n) = \left(c-2\times n\right)\times\left(c-2\times n-1\right)$$

$$Q(n) = 1-\left(1-\frac{PO_{h(x)}}{r\times(r-1)}\right)^{n}$$

$$PO_{h(x)} = \sum_{a=1}^{r}\sum_{b=1}^{r}\left[h(a)\times h(b)\times\left(\frac{\prod_{i=0}^{a-1}\frac{r-b-i}{r-i}}{P_c}\right)\times a\times b\right]$$

Proof

This is just a restatement of Theorem 6.17 with the modifications necessary to account for a column density distribution h(x) as specified in Theorems

6.18 and 6.19.

Q.E.D.

In order to test the suitability of the above formulae for representing the effects of column density distribution on foldability, the following test was undertaken. Two sets of 200 random PLAs of size r = 50, c = 14 and density = 0.14 were generated. In the first set, S_1, the PLAs had evenly distributed column densities, with each column having exactly 7 (0.14 × 50) active intersections. These PLAs were generated by the random PLA generator found in the appendix. In the second set, S_2, half of the columns in each PLA were given a density of 0.06 (3 active intersections), and half of the columns a density of 0.22 (11 active intersections). This particular column density distribution is an arbitrary one chosen to illustrate the effects of non evenly distributed column densities. A modified version of the random PLA generator found in the appendix was used to generate the PLAs in S_2.

All of the PLAs were folded using the basic random selection heuristic and the # of folds achieved was averaged for each of the two sets. The results were as follows.

1) The basic random selection heuristic found an average of 3.3 folds per PLA for the PLAs in S_1.

2) The formula in Theorem 3.17 predicted an average of 3.5 folds per PLA for the evenly dense PLAs of S_1.

It is not surprising that the formula predicts slightly more folds than that which actually occurs in practice. This is because the formula slightly

underestimates the value of Q(n) by only considering primary row orders (see figure 6.8).

3) For the PLAs in S_2, with a column density distribution as described above, the formula of Theorem 6.20 predicts an average of 5.55 folds per PLA, representing a 56% increase in folds due to the effect of the distribution of the column density. Recall that the average density of the PLAs in S_1 and S_2 are identical.

4) For the PLAs in S_2, the basic random selection heuristic actually found an average of 5.04 folds per PLA, representing an actual increase of 52% in folds due to the effect of the column density distribution.

The above data indicates that the formula in theorem 6.20 reasonably models the effects of column density distribution on the foldability of random PLAs.

Chapter 7

The Extended Random Selection Heuristic

It may be desirable to obtain optimal folding sets when folding PLAs, but this can become prohibitively expensive for even moderately sized industrial PLAs. Thus, for pragmatic reasons, one is forced to use heuristic methods to find a near-optimal solution to the problem. This chapter presents a new heuristic for PLA folding. It is shown to perform significantly better than the other heuristics available in the literature. This heuristic is based on the random selection heuristic and is called the *Extended Random Selection Heuristic*.

One might think that the random selection heuristic may be suitable only for theoretical study and that it would never be used to actually fold PLAs. This is not true. Empirical data presented in this chapter indicates that, in some cases, heuristics described in the literature do not perform much better than the simple random selection heuristic of the previous chapter. In addition, this new heuristic is the first PLA folding heuristic to have an analytical basis. Using this basis, one can predict the expected results of the algorithm for PLAs of a given size and density. Empirical results are included to demonstrate the effectiveness of the analytical derivation. The heuristic also can be adjusted to achieve results arbitrarily close to those of optimal folding. This allows one to adapt the algorithm to the amount of computing resources available and the nearness to optimality desired.

7.1. Analytical Derivation

Recall the basic random selection heuristic of the previous chapter.

Random Selection Heuristic

$F = \phi$; /* F is the folding set */

S = the set of all column pairs;

while ($S \neq \phi$) do

randomly select a pair p_1 from S;

$S = S - \{p_1\}$

If (the column pair p_1 is foldable including considerations of constraints imposed by previous folds in F) then

$F = F \bigcup p_1$

endif;

endwhile;

The extended form of the random selection heuristic is as follows.

Extended Random Selection Heuristic

$G = \phi$

for K times do

apply basic random selection to produce folding set F

if (cardinality of F) > (cardinality of G) then G = F

end for

K is called the *Repetition factor* of the heuristic.

Note that the time complexity of the algorithm is linear with respect to K and the basic random selection heuristic involves no backtracking of the search tree. Even for large PLAs, this represents a minimal amount of

computer time.

Theorem 7.1

The probability density function, $pdf_E(n)$, for expected number of folds using the Extended Random Selection Heuristic is given by

$$pdf_E(n) = \left[\sum_{i=0}^{n} pdf(i)\right]^K - \left[\sum_{i=0}^{n-1} pdf(i)\right]^K$$

where

K is the repetition factor of the heuristic as described above and

pdf(n) is the probability density function of the basic random selection heuristic, pdf(n,r,c,d), as derived in Theorem 6.17 of the last chapter. (Note that the parameters r,c and d are omitted for the sake of clarity. They will be assumed to be present when the term "pdf(n)" is used.)

Proof

The following proof employs the theory of order statistics which can be found in most intermediate statistics texts [GWH79].

The cumulative density function of pdf(n) is denoted Cdf(n) and is defined as

$$Cdf(n) = \sum_{i=0}^{n} pdf(i)$$

For a sample of K trials of the random selection heuristic, the cumulative density function of the ***largest*** resulting folding found is denoted as G(n) and is given by

$$G(n) = \left[Cdf(n)\right]^K$$

Let us denote g(n) as the probability density function of the largest folding found. If G(n) is the cumulative density function, then it can be seen that

g(n) will be the derivative of G(n). However, G(n) is not a continuous function but rather a discrete function only defined for integer values of n. Thus

$$\begin{aligned} g(n) &= G(n) - G(n-1) \\ &= \left[Cdf(n)\right]^K - \left[Cdf(n-1)\right]^K \\ &= \left[\sum_{i=0}^{n} pdf(i)\right]^K - \left[\sum_{i=0}^{n-1} pdf(i)\right]^K \end{aligned}$$

This is the pdf for expected number of folds of the order statistic heuristic with K repetitions. Thus,

$$pdf_E(n) = g(n) = \left[\sum_{i=0}^{n} pdf(i)\right]^K - \left[\sum_{i=0}^{n-1} pdf(i)\right]^K$$

Q.E.D.

Note that there is a slight error in Theorem 7.1. The formula for $pdf_E(n)$ does not remain valid for arbitrarily large values of K. There is an implicit assumption that the K trials of random selection are independent. This is not exactly true. The K trials are all performed on the same PLA. All of the trials are affected by any characteristic peculiar to that PLA. The fundamental characteristic, i.e # of rows, # of columns and density, are all accounted for by the formula, but there may be other more specific characteristics. As an example, the PLA may be one of the few on the fringe of the probability density function and be much more difficult to fold than its fundamental characteristics would indicate.

Figure 7.1 shows a plot of $pdf_E(n)$ for K=2 and K=10. It was derived from the formula of Theorem 7.1 with r=16, c=16 and d=25%. It can be

seen that by increasing the repetition factor K, the shape of the probability function remains relatively unchanged. It is however shifted, because higher K values result in more folds being found.

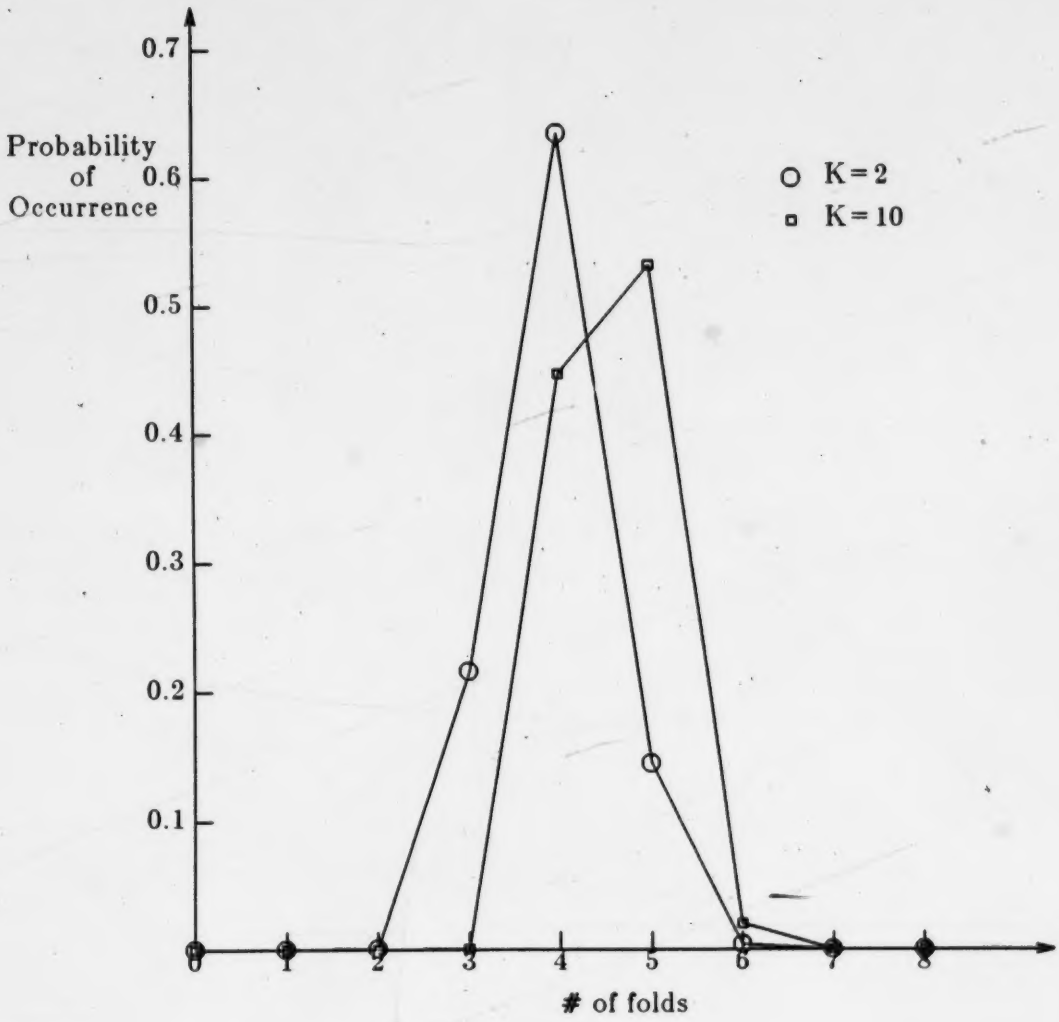

Figure 7.1 $pdf_E(n)$ for K=2 and K=10
for PLAs of size r=c=16 and density=25%

The amount that the probability density function shifts is best measured

by determining the average folding size expected for different K values. The average folding size is given by the follow formula.

$$average\ folding\ size = \sum_{i=0}^{C/2} \left(pdf_E(i) \times i \right)$$

Figure 7.2 show a plot of average folding size vs K for PLAs of size r= 16, c= 16 and density= 25%. Each point on the graph represents the average folding size as derived from $pdf_E(n)$ with the specified repetition factor. The graph appears to be asymtotically approaching some limit as K is increased.

7.2. Empirical Data

The Extended Random Selection Heuristic has been implemented and some empirical data have been produced. Figure 7.3 shows a plot of the probability density function of the Extended Random Selection Heuristic for K= 10, as derived both from the formula and as measured by applying the heuristic to 500 randomly generated PLAs. The PLAs were all of size r= 16, c= 16, density= 25% and were generated using the algorithm in the appendix..

Figure 7.4 shows a plot of average folding set size vs the repetition factor K, for the Extended Random Selection Heuristic, both as derived from the formula and as measured from a set of 500 randomly generated PLAs. The PLAs were all of size r= 16, c= 16 and density= 25%. Note that the plot for the measured values increases asymtotically in a manner identical to that of the derived value. However, there is a difference in the two plots in that the measured values are slightly lower than the corresponding derived values. This can be explained by recalling the assumptions under which the pdf function was derived in Chapter 6. The derived function ignores secondary

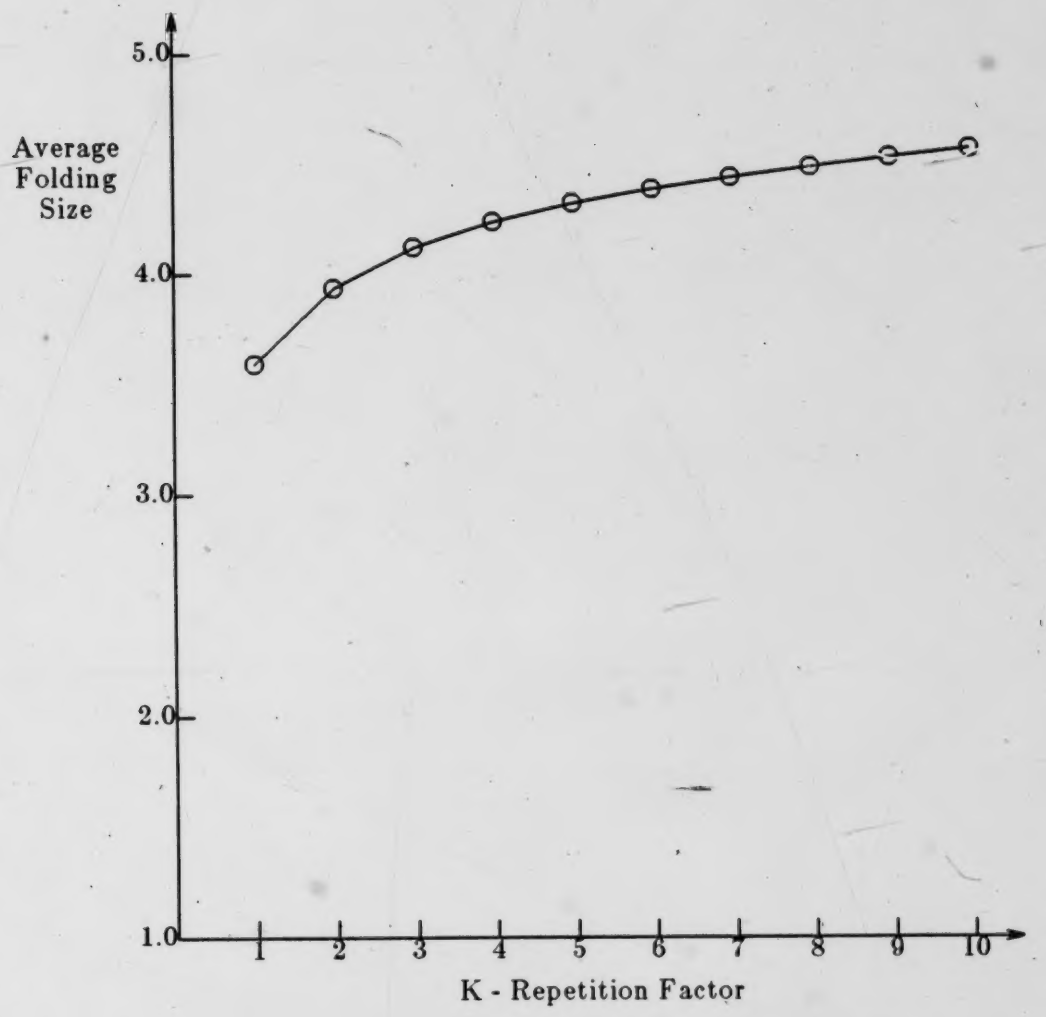

Figure 7.2 Average Folding Size vs K as derived from $pdf_E(n)$ for PLAs of size r = c = 16 and d = 25%

row order constraints which are not ignored when measuring real PLAs. Thus it is not surprising to find that the derived function estimates slightly more folding than that which occurs in real PLAs. As K increases in value, this difference is accentuated because the derived value is assuming independent random trials as was mentioned above. The real PLAs are

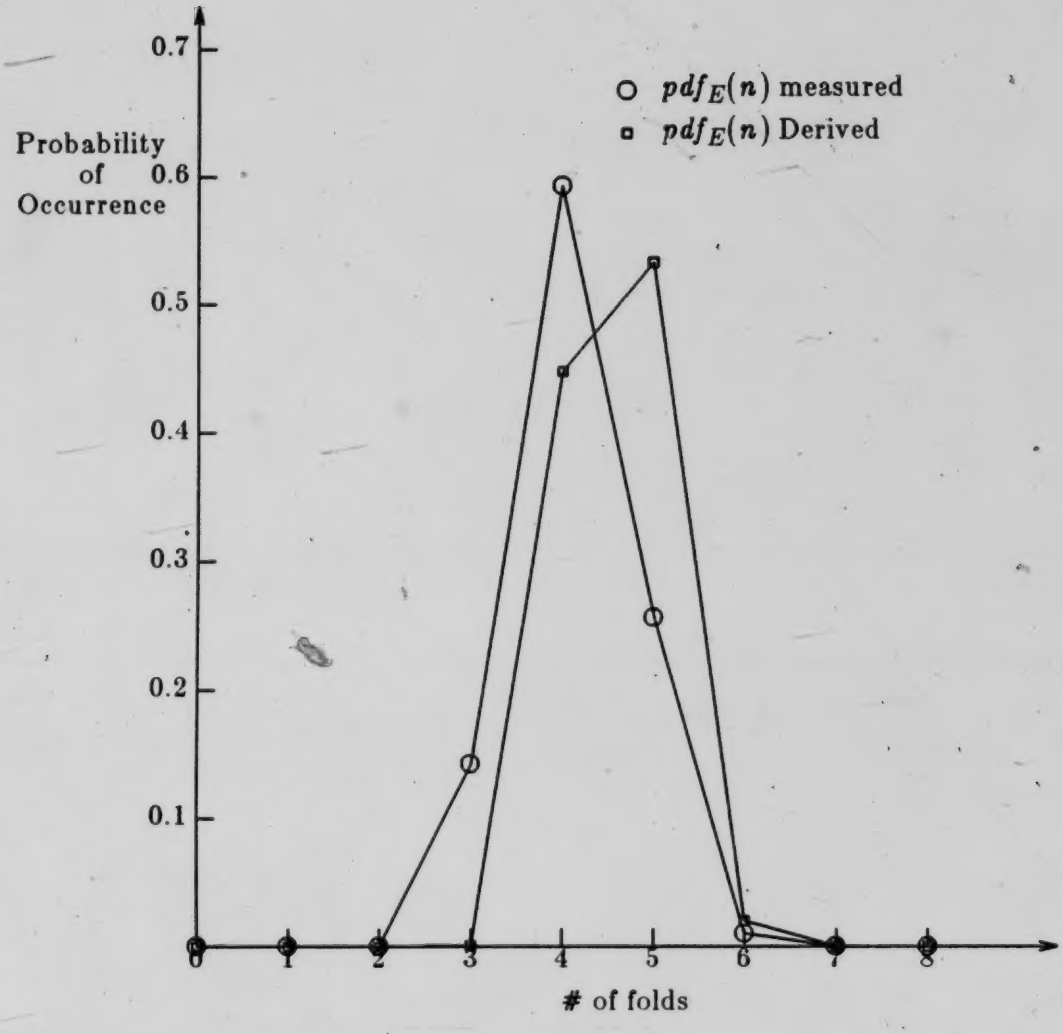

Figure 7.3 $pdf_E(n)$ for K = 10 as derived and measured for PLAs of size r = c = 16 and d = 25%

inherently, slightly more difficult to fold than the ones which the derived formula assumes to exist. The K different trials are all applied to this same set of real PLAs, whereas the derived formulas assumes that the K trials are performed on different, independent PLAs.

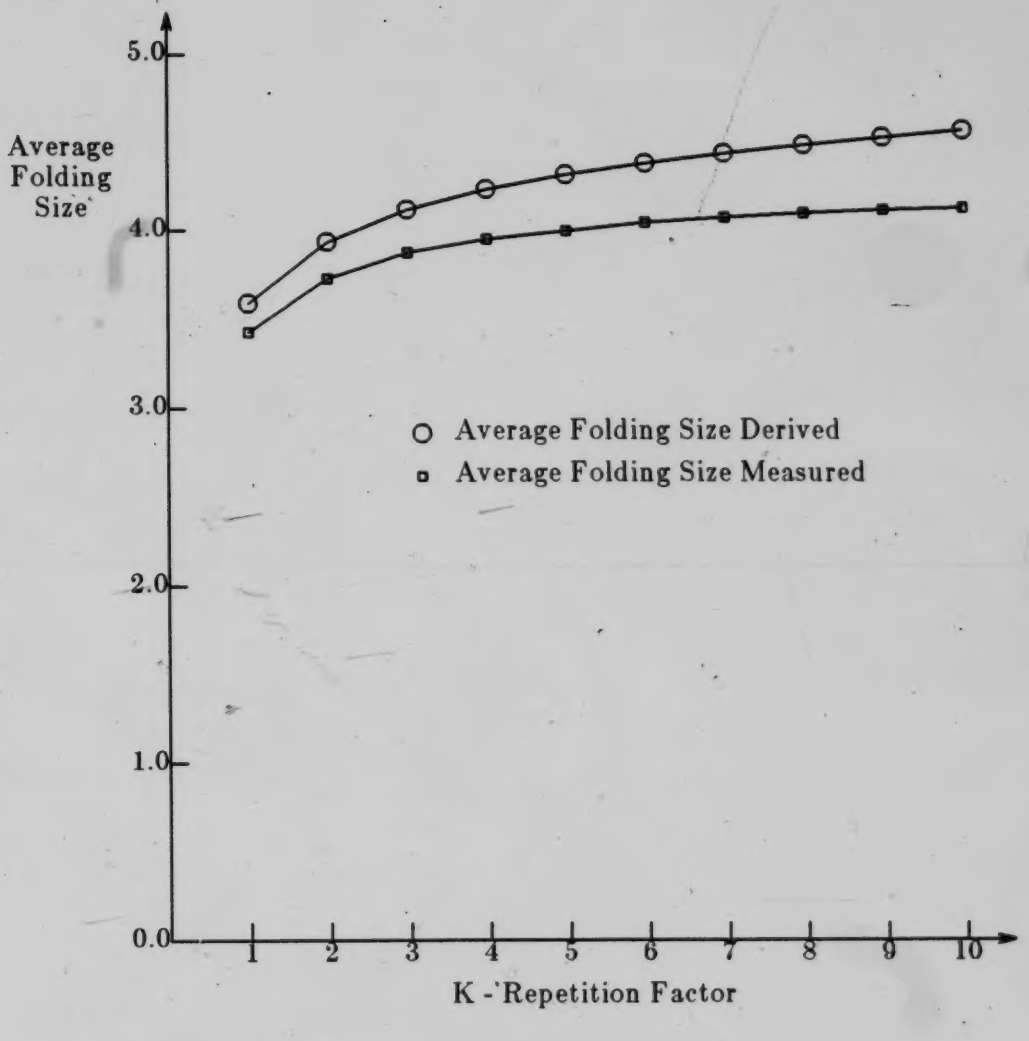

Figure 7.4 Average Folding Size vs K as derived and measured from $pdf_E(n)$ for PLAs of size r = c = 16 and d = 25%

These plots illustrate that the analytical derivation of the Extended Random Selection Heuristic appears to be reasonably effective in representing actual folding of random PLAs.

This new heuristic is the first PLA folding heuristic to have an analytical basis. In addition, the next chapter will show that, in a practical sense, it

outperforms other PLA folding heuristics found in the literature.

Chapter 8

Heuristic Comparison Measures

The current technique used in the literature for demonstrating the effectiveness of various heuristic PLA folding algorithms is to list the chip area savings achieved for a small number of specific PLAs [Gra82, HNS82a, HNS82b, KCH85, LeL84, LVV82, MiS83b] . This method is insufficient. Many PLAs may be difficult to fold despite the effectiveness of the folding heuristic. As well, many PLAs may produce large chip area savings despite the use of a poor heuristic.

To overcome this problem, this chapter presents two new measures for comparing the relative performance of heuristic PLA folding algorithms. These measures are statistically based and are defined for specific classes of PLAs.

For a given heuristic algorithm, the *Optimal Folding Ratio* is defined as the average ratio of

$$\frac{\textit{(folding set size using the heuristic)}}{\textit{(folding set size using optimal folding)}}$$

over a large random sample of PLAs of a given size and density. This indicates, on average, how close to optimal folding the heuristic achieves.

The *Heuristic Improvement Ratio* is defined as the average ratio of

$$\frac{\textit{(folding set size using given heuristic)}}{\textit{(folding set size with Basic Random Selection Heuristic (K = 1))}}$$

over a large random sample of PLAs of given size and density.

(The Basic Random Selection Heuristic is as defined in Chapter 6.) This

measure indicates, on average how much better the given heuristic performs when compared to basic random selection. If, for example, a new heuristic has a Heuristic Improvement Ratio of 1 of less, then it would be considered a very poor heuristic. One would expect a good heuristic to show a reasonable amount of improvement over basic random selection.

Table 8.1 show the results of a study done to compare the four column adjacency based heuristics popular in the literature [HNS82b] and the Extended Random Selection Heuristic presented earlier. All of these results were determined by using the various heuristics to fold 500 PLAs of size r = 16, c = 16 and density = 25%. The PLAs were generated using the algorithm in the appendix. It can been seen from the table that the Extended random Selection Heuristic performs better than any of the others. None of the other heuristics perform even adequately. Note the results for the widely used Min-Max heuristic. It has a heuristic improvement ratio of 1.065. This means that it only finds, on average, 6.5% more folds than even basic random selection. This contrasts to the Extended Random Selection Heuristic which finds 25% more folds than basic random selection. In fact, the Extended Random Selection Heuristic with K = 10 averages 95% of optimal folding results. This table clearly demonstrates that the Random Selection Heuristic is a good heuristic for practical PLA folding. The exact entries in the table would be different if PLAs of a different size and density were chosen, however the basic conclusions would be the same.

	Min-Min	Min-Max	Max-Min	Max-Max	Rand. Select. K=1	Rand. Select. K=10
Optimal Folding Ratio	.701	.819	.767	.738	.801	.957
Heuristic Improv. Ratio	.875	1.065	.995	.955	1.00	1.25

Table 8.1 Heuristic Comparison Measures

Chapter 9

Conclusions

The results of this research have been a number of fundamental contributions to the field of PLA folding. First, a set-theoretic model for the PLA folding problem was described. This new model has several advantages in its own right, and provides at least one alternative to the graph-theoretic model which is used almost exclusively in the literature. The thesis also provides a topological model for describing PLAs from a topological point of view. No other such model has been presented in the literature. As PLAs become more complex and diverse, this type of model will be required because it is flexible enough to describe multi-layer and multi-dimensional PLAs.

As an aid to understanding and comparing the many different PLA types and associative folding algorithms, a PLA taxonomy has been presented that is more complete than previous ones. This taxonomy classifies PLAs by both topological considerations and types of folding allowed. It also classifies the folding algorithms by their various properties. The complexity of this taxonomy illustrates the diversity of the PLA folding field.

A new branch and bound algorithm for optimal PLA folding has been developed. It has been shown to provide considerable savings in computer time in searching for optimal folding sets. To allow other researchers to compare the performance of branch and bound algorithms, two optimal folding measures have been described. The use of these measures eliminate the need to implement other researchers' algorithms for comparison purposes. The results of these measures are provided, as derived from the branch and

bound algorithm of this thesis.

The major contribution of this thesis is the theoretical analysis of the PLA folding problem found in Chapter 6. The analysis examines the fundamental parameters and properties of PLAs, how they restrict folding, and their interaction. The major result of this work is a probability density function for expected number of folds of random PLAs, defined in terms of the fundamental properties, r - number of rows, c - number of columns, d - density and h(x) the distribution of the density among the columns. The formula for the pdf is complex and represents the exact nature of the PLA folding problem. Empirical data have been provided to illustrate that the formulae reasonably model the actual folding of random PLAs.

The pdf formula was then used to develop a new heuristic for PLA folding. It is called the Extended Random Selection Heuristic and is the first heuristic to have an analytical basis for expected results. Empirical data are provided to illustrate its performance.

In demonstrating the effectiveness of this new heuristic, two new comparison measures were introduced. The use of these measures should eliminate the improper techniques now used to compare various heuristic algorithms. The results of these measures demonstrate that the Extended Random Selection Heuristic performs better than other heuristic methods previously discussed in the literature.

Together, the above contributions provide a much clearer understanding of the PLA folding problem than was previously available. However, there is work that remains to be done. For optimal PLA folding, branch and bound

algorithms should be developed which encompass all of the current bounding techniques. For heuristic folding, there still is not a good PLA folding heuristic available which is based on an understanding of the nature of PLA folding. The Extended Random Selection Heuristic presented here is a good one, but it is based on a random selection method. One would expect that it could be improved upon. The effect of secondary partial orders should be accounted for either analytically, if possible, or else through some empirical study. This would make the formulae more accurate. The properties of PLAs in industrial applications need to be studied to find out how they differ from those of random PLAs.

Finally, all of the work presented in this thesis needs to be expanded to encompass other types of PLA folding rather than limited to simple column folding.

References

[ArB78] Zosimo Areralo and Jon C. Bredeson, A Method to Simplify a Boolean Function into a Near Minimal Sum-of-Products for PLAs, *IEEE Transactions of Computers C-27*, 11 (Nov. 78), .

[Ayr83] Ronald F. Ayres, *VLSI: Silicon Compilation and The Art Of Automatic Microchip Design*, Prentice Hall, New Jersey, 1983.

[Bro81] Douglas W. Brown, A State-Machine Synthesizer, *18th Design Automation Conference*, 1981, pp301-305.

[EgL82] J. R. Egan and C. L. Liu, Optimal Bipartite Folding of PLA, *19th Design Automation Conference*, 1982, pp141-146.

[EgL84] J. R. Egan and C. L. Liu, Bipartite Folding and Partitioning of a PLA, *IEEE Transactions on Computer-Aided Design of I.C. Circuits and Systems CAD-3* , 3 (July 1984), pp191-199.

[FlM75] H. Fleisher and L.I. Maissel, An Introduction to Array Logic, *IBM Journal of Research and Development 19*, 2 (March, 1975), pp98-109.

[Goa81] Gary B. Goates, ABLE: A Lisp-Based Layout Modeling Language with User-Definable, *18th Design Automation Conference*, 1981, pp322-329.

[GLL80] R. L. Golden, P.A. Latos and P. Lowy, Design Automation and the PLA Macro, *IBM Journal of Research and Development 24*, 1 (Jan. 1980), pp23-31.

[GrN83] A. C. de Graaf and R. Nouta, Layout Generation of PLA Based Circuits from a Register Transfer Description, *IEEE International Conference on Computer Design, VLSI in Computers (ICCD 83)*, 1983, pp519-522.

[Gra82] Werner Grass, A Depth-First Branch-And-Bound Algorithm for Optimal PLA Folding, *19th Design Automation Conference*, 1982, pp133-140.

[GWH79] Irwin Guttman, S.S. Wilks and J. Stuart Hunter, *Introductory Engineering Statistics*, John Wiley and Sons, 1979.

[HSN80] G. D. Hachtel, A. Sangiovanni-Vincentelli and A. R. Newton, Some Results in Optimal Pla Folding, *IEEE International Conference on Circuits and Computers (ICCC 80)*, 1980, pp1023-1027.

[HNS82a] G. D. Hachtel, A. R. Newton and A. Sangiovanni-Vincentelli, Techniques for Programmable Logic Array Folding, *19th Design Automation Conference*, 1982, pp147-155.

[HNS82b] G. D. Hachtel, A. R. Newton and A. Sangiovanni-Vincentelli, An Algorithm for Optimal PLA Folding, *IEEE Transactions on Computer-Aided Design of I.C. Circuits and Systems CAD-1* , 2 (April 1982), .

[Hen83] John Hennessy, Partitioning Programmable Logic Arrays Summary, *International Conference on Computer-Aided-Design (ICCAD 83)*, 1983, pp180-181.

[Hia84] Fredrick Hill and et al, Hardware Compilation from an RTL to a Storage Logic Array Target, *IEEE Transactions on Computer-Aided Design of I.C. Circuits and Systems CAD-3* , 3 (July 1984), pp208-217.

[HCO74] S.J. Hong, R.G. Cain and D. L. Ostapko, Mini: A Heuristic Approach for Logic Minimization, *IBM Journal of Research and Development 18*, 5 (Sept, 1974), pp443-458.

[Hu83] T. C. Hu and Y. S. Kuo, Optimum Reduction of Programmable Logic Array, *20th Design Automation Conference*, 1983, pp553-558.

[Jon75] J. W. Jones, Array Logic Macros, *IBM Journal of Research and Development 19*, 2 (March, 1975), pp120-126.

[Kam79] Yahiko Kambayashi, Logic Design of PLAs, *IEEE Transactions of Computers C-28*, 9 (Sept. 79), pp609-617.

[KaC81] S. Kang and W. M. van Cleemput, Automatic PLA Synthesis from a DDL-P Description, *18th Design Automation Conference*, 1981, pp391-397.

[Kan81] Sungho Kang, Synthesis and Optimization of Programmable Logic Arrays, *PhD dissertation, Stanford University*, 1981.

[KCH85] Y. S. Kuo, C. Chen and T. C. Hu, A Heuristic Algorithm for PLA Block Folding, *22nd Design Automation Conference*, 1985, pp744-747.

[LeL84] J. L. Lewandowski and C. L. Liu, A Branch and Bound Algorithm for Optimal PLA Folding, *21st Design Automation Conference*, 1984, pp426-433.

[Loa75] J. C. Logue and et al, Hardware Implementation of a Small System in PLAs, *IBM Journal of Research and Development 19*, 2 (March, 1975), pp110-119.

[LVV82] M. Luby, U. Vazirani, V. Vazirani and A. Sangiovanni-Vincetelli, Some Theoretical Results on the Optimal PLA Folding Problem, *IEEE International Conference on Circuits and Computers (ICCC 82)*, 1982, pp165-170.

[LTM82] W. Luciw, J. B. Tomei and K. D. Mandl, A New Storage Logic Array (SLA) Cell for High Density VLSI, *Custom Integrated Circuits Conference*, 1982, pp120-124.

[MaT85a] D. Makarenko and J. Tartar, The Effect of Density on the Folding of Programmable Logic Arrays, *Canadian Conference on Very Large Scale Integration*, 1985.

[MaT85b] D. Makarenko and J. Tartar, An Efficient Algorithm for the Optimal Folding of PLAs, *IEEE International Conference of Computer Design: VLSI in Computers (ICCD'85)*, 1985.

[MaT86] D. Makarenko and J. Tartar, A Statistical Analysis of PLA Folding, *(to be published) IEEE Transactions on Computer-Aided Design of I.C. Circuits and Systems*, January, 1986.

[MaP83] Jorge F. Martinez-Carballido and V.Michael Powers, Pronto: Quick PLA Product Reduction, *20th Design Automation Conference*, 1983, pp545-552.

[MeC80] Carver Mead and Lynn Conway, *Introduction to VLSI Systems*, Addison-Wesley Publishing Company, 1980.

[MAP84] M. J. Meyer, P. Agrawal and R. G. Pfister, A VLSI FSM Design System, *21st Design Automation Conference*, 1984, pp434-440.

[MiS83a] Giovanni De Micheli and Mauro Santomauro, Topological Partitioning of Programmable Logic Arrays, *International Conference on Computer-Aided-Design (ICCAD 83)*, 1983, pp182-183.

[MiS83b] Giovanni De Micheli and A. Sangiovanni-Vincentelli, Pleasure: A Computer Program for Simple/Multiple Constrained/Unconstrained Folding of Programmable Logic Arrays, *20th Design Automation Conference*, 1983, pp530-537.

[MiS83c] Giovanni De Micheli and A. Sangiovanni-Vincentelli, Multiple Folding of Programmable Logic Arrays, *IEEE International Symposium on Circuits and Systems*, 1983, pp1026-1029.

[MSV83] Giovanni De Micheli, Alberto Snagiovanni-Vincentelli and Tiziano Villa, Computer-Aided Synthesis of PLA-BAsed Finite State Machines, *International Conference on Computer-Aided-Design (ICCAD 83)*, 1983, pp154-156.

[MiS83] Giovanni De Micheli and A. Sangiovanni-Vincentelli, Multiple Constrained Folding of PLAs: Theory and Application, *IEEE Transactions on Computer-Aided Design of I.C. Circuits and Systems CAD-2*, 3 (July 1983), pp151-167.

[Pai81] J. F. Paillotin, Optimization of the PLA Area, *18th Design Automation Conference*, 1981, pp406-410.

[Pat80] Suhas S. Patil, On Testability of Digital Systems Designed with Storage/Logic Arrays, *IEEE International Conference on Circuits and Computers (ICCC 80)*, 1980, pp533-537.

[PaW79] Suhas S. Patil and Terry A. Welch, A Programmable Logic Approach for VLSI, *IEEE Transactions of Computers C-28*, 9 (Sept. 79), pp594-601.

[PuL84] Virapan Pulges and M. R. Lightner, A Recursive Bipartitioning Approach for PLA Partitioning and Layout, *IEEE International Conference on Computer Design, VLSI in Computers (ICCD 84)*, 1984, pp320-323.

[RaL84] S.S. Ravi and E.L. Lloyd, The Complexity of Near-Optimal PLA Folding, Technical Report 84-18, Department of Computer Science, State University of New York at Albany , December 1984.

[Rot78] . Roth, PLA Optimization, *IEEE Transactions of Computers C-27*, 2 (Feb. 78), .

[San85] A. Sangiovanni-Vincentelli, An Overview of Synthesis Systems, *IEEE Custom Integrated Circuits Conference (CICC 85)*, 1985, pp221-225.

[Sas84] Tsumtomu Sasao, Input Variable Assignment and Output Phase Optimizationof PLAs, *IEEE Transactions of Computers C-33*, 10 (Oct. 84), pp879-894.

[Sas81] Tsumtomu Sasao, Multi Valued Decomposition of Generalized Boolean Functions and the Complexity of PLAs, *IEEE Transactions of Computers C-30*, 9 (Sept. 81), pp635-643.

[Sch80] Martin S. Schmookler, Design of Large ALUs Using Multiple PLA Macros, *IBM Journal of Research and Development 24*, 1 (Jan. 1980), pp2-14.

[Smi82] Kent Farrell Smith, Design of Integrated Circuits with Structured Logic Using the Storage/Logic Array (SLA) Definition and Implementation, *PHd dissertation, The University of Utah*, 1982.

[Sta82] M. W. Stebnisky and et al., A Fully Automatic, Technology-Independent PLA Macrocell Generator, *IEEE International Conference on Circuits and Computers (ICCC 82)*, 1982, pp156-160.

[Sta83] M. W. Stebnisky and et al., APSS: An Automatic PLA Synthesis System, *20th Design Automation Conference*, 1983, pp430-435.

[SGM83] Fabio Sumenzi , Silvano Gai, Marco Mezzalama and Paolo Prinetto, A New Integrated System fo PLA Testing and Verfication, *20th Design Automation Conference*, 1983, pp57-63.

[SuK81] I. Suwa and W. J. Kubitz, A Computer-Aided-Design System for Segmented-Folded PLA Macro-Cells, *18th Design Automation Conference*, 1981, pp398-405.

[Suw82] Izumi Suwa, Topological Design of Segmented-Folded PLAs, *PHd dissertation, University of Illinois-Urbana*, 1982.

[TeW82] Bill Teel and Doran Wilde, A Logic Minimizer for VLSI PLA Design, *19th Design Automation Conference*, 1982, pp156-162.

[TFB82] C. C. Timoc, J. M. Favennec and C. Le Blanche, A Testable Regular Design, *IEEE International Conference on Circuits and Computers (ICCC 82)*, 1982, pp210-213.

[Wei79] Arnold Weinberger, High Speed PLA Adders, *IBM Journal of Research and Development 23*, 2 (March, 1979), pp163-178.

[WiS83] S. Wimer and N. Sharfam, HOPLA - PLA Optimization and Synthesis, *20th Design Automation Conference*, 1983, pp790-794.

[Woo79] Roy A. Wood, A High Density Programmable Logic Array Chip, *IEEE Transactions of Computers C-28*, 9 (Sept. 79), .

Appendix A1

Random PLA Generator

The following is an outline of the program used to generate random PLAs with even column density. This means that each column of the PLA will have the same number of active intersections. The function uniform(n) is a random number generator which produces a random number from 1 to n with a uniform distribution. pla[ncols][nrows] is an array used to store the generated PLA with a 1 indicating an active intersection and a 0 indicating otherwise.

```
plagen(nrows,ncols,dens) {
  /*
    nrows is the number of rows of the PLA
    ncols is the number of columns of the PLA
    dens is the density of the PLA
  */
  int uniform();
  int pla[ncols][nrows];

  for(i=1;i<=ncols;i++) { /* loop through all columns */

    /* calc # of active intersections initially
      left to place in this column */
    activeleft=dens*nrows;

    for(j=1;j<=nrows;j++ { /* loop through all rows */
      if(uniform(nrows-i+1) <= activeleft)
        { pla[i][j]=1;
          activeleft--;
        }
      else pla[i][j]=0;
    } /* end for nrows */
  } /* end for columns */
} /* end or plagen */
```